

H-A.

0400  
0400

0500  
0280

500.39846X00

#4

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): M. YOSHIMURA, ET AL.

Serial No.: 09 / 803,149

Filed: MARCH 12, 2001

Title: "MULTI-DATABASE SYSTEM".

LETTER CLAIMING RIGHT OF PRIORITY

Honorable Commissioner of  
Patents and Trademarks  
Washington, D.C. 20231

MARCH 29, 2001

Sir:

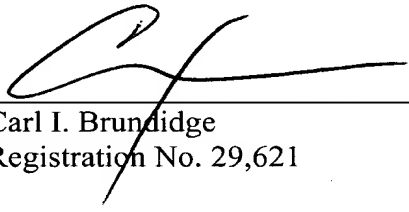
Under the provisions of 35 USC 119 and 37 CFR 1.55, the applicant(s) hereby claim(s)  
the right of priority based on:

Japanese Patent Application No. 2000 - 342373  
Filed: NOVEMBER 6, 2000

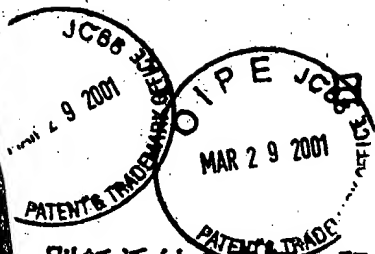
A certified copy of said Japanese Patent Application is attached.

Respectfully submitted,

ANTONELLI, TERRY, STOUT & KRAUS, LLP

  
\_\_\_\_\_  
Carl I. Brundidge  
Registration No. 29,621

CIB/rp  
Attachment



本 国 特 許 庁  
PATENT OFFICE  
JAPANESE GOVERNMENT

別紙添付の書類に記載されている事項は下記の出願書類に記載されて  
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed  
with this Office.

出 願 年 月 日  
Date of Application:

2000年11月 6日

出 願 番 号  
Application Number:

特願2000-342373

願 人  
Applicant(s):

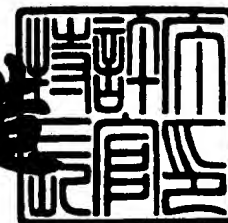
株式会社日立製作所

CERTIFIED COPY OF  
PRIORITY DOCUMENT

2001年 3月16日

特許庁長官  
Commissioner,  
Patent Office

及 川 耕 造



出証番号 出証特2001-3018514

【書類名】 特許願

【整理番号】 K00010311

【提出日】 平成12年11月 6日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 17/30

【請求項の数】 10

【発明者】

    【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

    【氏名】 吉村 光彦

【発明者】

    【住所又は居所】 神奈川県川崎市麻生区王禅寺 1 0 9 9 番地 株式会社日立製作所 システム開発研究所内

    【氏名】 矢川 雄一

【発明者】

    【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

    【氏名】 林 重年

【発明者】

    【住所又は居所】 神奈川県横浜市戸塚区戸塚町 5 0 3 0 番地 株式会社日立製作所 ソフトウェア事業部内

    【氏名】 西川 記史

【特許出願人】

    【識別番号】 000005108

    【氏名又は名称】 株式会社日立製作所

【代理人】

    【識別番号】 100075096

    【弁理士】

    【氏名又は名称】 作田 康夫

【手数料の表示】

【予納台帳番号】 013088

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【ブルーフの要否】 要

【書類名】 明細書

【発明の名称】 マルチデータベース統合システムのアクセス権管理方法

【特許請求の範囲】

【請求項 1】

アプリケーションを実行するクライアントおよびそれぞれがデータベースを有する少なくとも1つのデータベースサーバが接続されたマルチデータベース統合サーバは、

前記クライアントから、前記アプリケーションで使用される仮想表の仮想表定義情報に含まれる操作種別とユーザ名を定めた仮想表アクセス権情報とを入力し

、  
前記データベースサーバのそれぞれから、接続ユーザ名と表操作権限とを対応付けたデータベースアクセス権情報を入力し、

前記ユーザ名と前記接続ユーザ名、および前記操作種別と前記表操作権限とをそれぞれ比較することによって、前記データベースのそれぞれに対する前記仮想表のアクセス権を判定することを特徴とするマルチデータベース統合システムのアクセス権管理方法。

【請求項 2】

前記アクセス権管理方法は、さらに、前記アクセス権の判定結果に基づいて、前記データベースのそれぞれに対する前記仮想表のアクセス権の有無を前記クライアントの表示装置に表示することを特徴とする請求項 1 記載のマルチデータベース統合システムのアクセス権管理方法。

【請求項 3】

アプリケーションを実行するクライアントおよびそれぞれがデータベースを有する少なくとも1つのデータベースサーバが接続されたマルチデータベース統合サーバは、

前記クライアントから、前記アプリケーションで使用される仮想表の仮想表定義情報に含まれる操作種別とユーザ名を定めた仮想表アクセス権情報とを入力し

、  
前記データベースサーバのそれぞれから、接続ユーザ名と表操作権限とを対応

付けたデータベースアクセス権情報を入力し、

前記仮想表からのアクセスの際に、前記操作種別と前記表操作権限との整合性をチェックした上で、前記ユーザ名を前記接続ユーザ名に変換することによって、前記データベースのそれぞれに対する前記仮想表のアクセス権を変換することを特徴とするマルチデータベース統合システムのアクセス権管理方法。

【請求項 4】

前記アクセス権を変換するステップにおいて、所定のルールに基づいて前記アクセス権を変換することを特徴とする請求項 3 記載のマルチデータベース統合システムのアクセス権管理方法。

【請求項 5】

アプリケーションを実行するクライアントおよびそれぞれがデータベースを有する少なくとも 1 つのデータベースサーバが接続されたマルチデータベース統合サーバは、

前記クライアントから、前記アプリケーションで使用される仮想表の仮想表定義情報に含まれる操作種別とユーザ名を定めた仮想表アクセス権情報とを入力し、

前記データベースサーバのそれぞれから、接続ユーザ名と表操作権限とを対応付けたデータベースアクセス権情報を入力し、

前記ユーザ名と前記接続ユーザ名、および前記操作種別と前記表操作権限とをそれぞれ比較することによって、前記データベースのそれぞれに対する前記仮想表のアクセス権を判定し、

前記判定の結果、アクセス権の不整合が判明したデータベースに対して、前記表操作権限が同一の他の接続ユーザ名によって前記接続ユーザ名を修復することの特徴とするマルチデータベース統合システムのアクセス権管理方法。

【請求項 6】

アプリケーションを実行するクライアントおよびそれぞれがデータベースを有する少なくとも 1 つのデータベースサーバが接続されたマルチデータベース統合サーバは、

前記クライアントから、前記アプリケーションで使用される仮想表の仮想表定

義情報に含まれるデータ項目毎の操作種別とユーザ名を定めた仮想表アクセス権情報とを入力し、

前記データベースサーバのそれぞれから、接続ユーザ名とデータ項目毎の表操作権限とを対応付けたデータベースアクセス権情報を入力し、

前記ユーザ名と前記接続ユーザ名、およびデータ項目毎の前記操作種別と前記表操作権限とをそれぞれ比較することによって、前記データベースのそれぞれに対する前記仮想表のアクセス権をデータ項目毎に判定することを特徴とするマルチデータベース統合システムのアクセス権管理方法。

【請求項 7】

前記アクセス権管理方法は、さらに、前記アクセス権の判定結果に基づいて、前記データベースのそれぞれに対する前記仮想表のアクセス権の有無をデータ項目毎に前記クライアントの表示装置に表示することを特徴とする請求項 6 記載のマルチデータベース統合システムのアクセス権管理方法。

【請求項 8】

アプリケーションを実行するクライアントおよびそれぞれがデータベースを有する少なくとも 1 つのデータベースサーバが接続されたマルチデータベース統合サーバは、

前記クライアントから、前記アプリケーションで使用される仮想表の仮想表定義情報に含まれる操作種別とユーザ名を定めた仮想表アクセス権情報とを入力し、

前記データベースサーバのそれぞれから、接続ユーザ名と表操作権限とを対応付けたデータベースアクセス権情報を入力し、

前記ユーザ名と前記接続ユーザ名、および前記操作種別と前記表操作権限とをそれぞれ比較することによって、前記データベースのそれぞれに対する前記仮想表のアクセス権を判定し、

前記仮想表からのアクセスの際に、前記アクセス権の判定結果に基づいて、前記データベースのそれぞれに対する前記仮想表のアクセス権を変換することを特徴とするマルチデータベース統合システムのアクセス権管理方法。

【請求項 9】

アプリケーションを実行するクライアントおよびそれぞれがデータベースを有する少なくとも 1 つのデータベースサーバが接続されたマルチデータベース統合サーバが実行するマルチデータベース統合システムのアクセス権管理方法のプログラムを格納した、計算機で読み取り可能な記憶媒体であって、前記アクセス権管理方法は、

前記クライアントから、前記アプリケーションで使用される仮想表の仮想表定義情報に含まれる操作種別とユーザ名を定めた仮想表アクセス権情報とを入力し

前記データベースサーバのそれぞれから、接続ユーザ名と表操作権限とを対応付けたデータベースアクセス権情報を入力し、

前記ユーザ名と前記接続ユーザ名、および前記操作種別と前記表操作権限とをそれぞれ比較することによって、前記データベースのそれぞれに対する前記仮想表のアクセス権を判定し、

前記仮想表からのアクセスの際に、前記アクセス権の判定結果に基づいて、前記データベースのそれぞれに対する前記仮想表のアクセス権を変換すること、を特徴とする記憶媒体。

【請求項 10】

前記アクセス権管理方法において、前記接続ユーザ名は、前記仮想表、ユーザ、および固定ユーザ名のいずれかと対応させることを特徴とする請求項 1, 3, 5, 6 および 8 に記載のマルチデータベース統合システムのアクセス権管理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、マルチデータベース統合システムにおけるアクセス権管理方法に関する。



## 【 0 0 0 2 】

## 【従来の技術】

従来技術として、複数のデータベース管理システム（以下、DBMSと略す）を一つのデータベースに統合するマルチデータベース統合システムが知られている。例えば、特開平 8 - 1 6 4 3 9 「異なるデータベースシステム間でのデータ共有装置及び方法」に述べられている。こうしたマルチデータベース統合システムを利用することにより、複数DBMSを利用したアプリケーション（以下APと略す）を容易に開発できる。マルチデータベース統合システムに対してのみ検索処理を行なうことにより、複数DBMSからの検索結果を取得できる。こうしたマルチデータベース統合システムでは、実際のデータ取得を統合対象となる複数データベースから得るために、統合対象となるデータベースに対するアクセス制御を行なう必要がある。

## 【 0 0 0 3 】

従来のマルチデータベース統合システムでは、マルチデータベース統合システムに対するユーザ名を、統合対象となるデータベースのユーザ名に変換することにより、統合対象データベースとの接続および表へのアクセス制御を行なっている。この際、マルチデータベース統合システムに対するアクセス権が成立するか否かは、変換されたユーザ名がおのこの統合対象DBMSにてアクセスできるか否かに基づいて行なわれる。International Standardization Organization（以下ISOと略す）で定めるDatabase LanguageSQLのSQL/MEDの規格では、マルチデータベース統合システムに対するユーザ名を、統合対象となるデータベースのユーザ名に対応させるマッピング定義について述べられている。

## 【 0 0 0 4 】

## 【発明が解決しようとする課題】

システム構築を行なう際、マルチデータベース統合システム上にて構築されるAP側と統合対象DBMS側の管理組織が異なる場合がある。こうした場合、AP側と統合対象DBMS間の接続を行なうには、それぞれ独立した形態でシステム構築作業を進められることが望ましい。しかし従来システムでは、マルチデータベース統合システムに対するユーザ名を、統合対象DBMS上のユーザ名に直接対応づけるため

、統合対象ユーザ統合対象側DBMSのアクセス権が定義できない限り、AP側の利用形態に応じたアクセス権定義がおこなえない。このため、AP側と統合対象DBMSのシステム構築作業を独立にすすめることができず、効率的なシステム開発を行なえない場合がある。

【0005】

また、従来のアクセス権管理方法では、統合対象DBMS側のログ情報を用いて、AP側のユーザを区別するための設定が煩雑な場合がある。例えば、「同一組織内のAPからマルチデータベース統合システム上を経由して統合対象DBMSを利用するユーザ」と、「他組織のAPからマルチデータベース統合システムを経由して統合対象DBMSを利用するユーザ」を区別するための設定を行なうには、AP側と統合対象DBMSのユーザ設計を同時に行なう必要がある。こうした作業の煩雑さのため、上記区別を行なうための設定が行なわれず、このためスポークDB側でセキュリティ上の問題が発生した場合には、捜査対象となるユーザグループの分類を行なえない場合がある。

【0006】

また、マルチデータベース統合システム上にて構築されるAP側と統合対象DBMS側の管理組織が異なる場合には、統合対象DBMS上のユーザ名に対して表操作権限の変更や、ユーザ名そのものの削除などが、AP利用者に通知されないまま、行なわれることがあるが、従来技術では、こうした際に生じる不整合を検出し、修復する手段をもっていない。

【0007】

本発明の目的は、AP側と結合対象DBMSのシステム構築を独立に進めることができるマルチデータベース統合システムのアクセス権管理方法を提供することにある。

【0008】

【課題を解決するための手段】

本発明は、マルチデータベース統合部の定義情報を定義する定義部、定義部の定義情報を格納する定義DB、複数データベースの統合を行なうマルチデータベース統合部よりなる。

## 【 0 0 0 9 】

定義部は、アクセス権マッピング定義部と、定義時アクセス権マッピングの整合性チェック部よりなる。アクセス権マッピング定義部は、マルチデータベース統合部上にて管理する仮想表に対するアクセス権と、統合対象DBMS上の実表に対するアクセス権とを対応づけ、対応関係を保存する処理を行なう。定義時アクセス権マッピングの整合性チェック部は、上記で定義した対応づけ関係に基づいて仮想表に対するアクセス権と、実表に対するアクセス権間で不整合が生じていないかを判定し、不整合を検出した場合には定義処理をおこなっているユーザにその旨を通知し、不整合の修復を支援する。

## 【 0 0 1 0 】

定義DBは、DBMS情報、ユーザ情報、仮想表定義、仮想表アクセス権、アクセス権マッピングを保持する。DBMS情報では、統合対象DBMSの管理情報を保持する。ユーザ情報では、仮想表ユーザを管理する。仮想表定義では、仮想表の定義を格納する。仮想表アクセス権では、仮想表に対するアクセス権を管理する。アクセス権マッピングでは、仮想表アクセス権と統合対象DBMSのアクセス権との対応関係を保持する。

## 【 0 0 1 1 】

マルチデータベース統合部は、実行時アクセス権マッピングの整合性チェック部、仮想表アクセス権判定部、アクセス権変換部よりなる。実行時アクセス権マッピングの整合性チェック部では、実行時の環境において、定義したアクセス権マッピングに不整合がないかを検出し、不整合を検出した場合には可能な範囲での修復処理を行なう。仮想表アクセス権判定部では、AP側で利用する仮想表に対するアクセス権判定を行なう。アクセス権変換部では、仮想表アクセス権を統合対象となるDBMSへのアクセス権に変換する処理を行なう。

## 【 0 0 1 2 】

## 【発明の実施の形態】

本実施例のアクセス権制御システムは、マルチデータベース統合システムにおける仮想表へのアクセス権制御を実現するシステムである。ここでいうマルチデータベースシステムとは、複数のデータベース管理システム（以下DBMSと略す）

に格納した表を組み合わせ、仮想表と呼ぶ一つの表を定義し、システム利用者がこの仮想表に対してDBMSの操作言語であるSQLなどを用いることで、レコード情報の検索（SELECT）、更新（UPDATE）、削除（DELETE）、新規作成（INSERT）などの問い合わせ処理を行なうシステムのことである。従来のDBMSでは、実データを格納している表に対して、複数の表を仮想的に組み合わせるビューとよばれる概念がある。マルチデータベースシステムの仮想表は、異なるDBMS上の表に対するビューとして定義する。更にマルチデータベースシステムでは、上述した仮想表に対して、従来のDBMSと同様に仮想表に対する操作権限をもつユーザや表操作権限の種別を対応づけ、仮想表に対するアクセス権情報の管理を行なう。なお、ここでいうアクセス権情報とは、表、ビュー、および仮想表、に対して、DBMSもしくはマルチデータベースシステムのユーザが、どのようなSQL操作種別（SELECT、UPDATE、DELETE、INSERT）を行なえばよいかを対応づけた情報のことである。特に仮想表に対するアクセス権情報のことを、仮想表アクセス権と呼ぶことにする。そして統合対象のDBMS上の表およびビューに対するアクセス権をDBMSアクセス権とよぶことにする。

## 【 0 0 1 3 】

以下では、図 1 を用いて本発明におけるマルチデータベース統合システムの概略動作説明を行う。

## 【 0 0 1 4 】

図 1 の定義部 1 0 では、定義DB20に対して、アクセス権マッピング 2 5 などの定義情報を登録する。登録するにあたっては、アクセス権マッピング 2 5 に不整合がないかのチェックを行う。

## 【 0 0 1 5 】

図 1 のマルチデータベース統合部 3 0 では、実行時のデータベース統合処理を行なう。図 1 のAP 6 0 は仮想表を利用したアプリケーションを示している。AP 6 0 からマルチデータベース統合部 3 0 に対して、仮想表に対する操作を行うに当たっては、まずユーザ名、パスワードを用いたログイン処理を行う。図 1 ではAP User1というユーザ名を用いて、AP 6 0 とマルチデータベース統合部 3 0 の間で接続処理を行っている。マルチデータベース統合部 3 0 では、定義DB 2.0 のユー

ザ情報 2 2 に、ユーザ名 APUser 1 およびパスワードが存在するか否かを確認し、ユーザ情報 2 2 に該当ユーザが存在した場合に、AP 6 0 とマルチデータベース統合部 3 0 間の接続を確立する。

#### 【 0 0 1 6 】

上記接続が確立した後、AP 6 0 からはマルチデータベース統合部 3 0 に対して仮想表に対する表操作処理を要求する。表操作処理の要求には、通常の DBMS 操作言語である SQL を用いて行う。例えば、仮想表 VT1 に対する SQL として「SELECT \* FROM VT1」を要求する。AP 6 0 からの前記 SQL を受け付けたデータベース統合部 3 0 では、仮想表アクセス権判定 3 2 において、APUser1 に対して、仮想表 VT1 に対しする SELECT 操作権限が、与えられているか否かを判定する。この仮想表に対するアクセス権の判定処理には、定義 DB 2 0 の仮想表アクセス権 2 4 を用いる。図 1 の仮想表アクセス権 2 4 では、APUser1 の VT1 に対する SELECT、UPDATE 権限が定義されおり、前述した SQL に対しては、仮想表アクセス権が成立するものと判定する。

#### 【 0 0 1 7 】

仮想表アクセス権が成立した場合、仮想表を構成するための実データを統合対象 DBMS から取得するための前処理として、マルチデータベース統合部 3 0 と統合対象 DBMS 間の接続確立を行う必要がある。図 1 では、DBMS 1 (40)、DBMS 2 (50) が統合対象の DBMS になっている。ここで注意する点は、マルチデータベース統合部 3 0 と統合対象 DBMS 間の接続に必要となるユーザ名、パスワードは、AP 6 0 とマルチデータベース統合部 3 0 間の接続を確立するのに用いたユーザ名 APUser1 と異なることである。仮想表アクセス権変換 3 3 において、仮想表アクセス権を統合対象 DBMS のアクセス権に変換する処理を行なうことにより、マルチデータベース統合部 3 0 と統合対象 DBMS 間の接続に必要なユーザ名が取得する。仮想表アクセス権変換 3 3 を行うにあたっては、定義 DB 2 0 上アクセス権マッピング 2 5 を参照する。

#### 【 0 0 1 8 】

仮想表アクセス権変換 3 3 により、DBMS 接続に必要なユーザ名を取得した後は、マルチデータベース統合部 3 0 と統合対象 DBMS 間の接続を確立し、マルチデ

データベース統合部 30 が、個々の統合対象 DBMS に対して検索処理などの表操作処理を行い、個々の統合対象 DBMS からの得た結果を、マルチデータベースシステム側で統合し、アプリケーション側に応答する。

#### 【0019】

以上が、マルチデータ統合システム 30 の動作概略である。

#### 【0020】

本発明は特に上記処理の内アクセス権制御にかかわるものである。特に本発明では、アクセス権変換として、以下の 3 方式について説明を行なう。これらの方式の構成を図 4、図 5、図 6 に示す。

#### 【0021】

(1) 仮想表単位に統合対象 DBMS 上のユーザ名を対応づける方式。図 4 の例では、仮想表 VT1 に対して統合対象 DBMS 上のユーザ名 DB1\_User1、DB2\_User1 を対応づける。

#### 【0022】

(2) マルチデータベースのユーザ単位に統合対象 DBMS 上のユーザ名を対応づける方式。図 5 の例では、マルチデータベースシステムのユーザ APUser1 に対して統合 DBMS 上のユーザ名 DB1\_User1、DB2\_User1 を対応づける。

#### 【0023】

(3) 統合対象 DBMS 単位に固定ユーザ名を対応づける方式。図 6 に示すように、統合 DBMS 単位に固定のユーザ名 DB1\_User1、DB2\_User1 を対応づける。

#### 【0024】

以下ではまず、上記 (1) の方式に基づく実施例 1 の形態を説明する。概略説明に用いた図 1 に対し、再度詳細な説明を行う。図 1 に示すように実施例 1 の処理ブロック図は、定義部 10、定義 DB 20、マルチデータベース統合部 30、DBMS1 40、DBMS2 50、AP 60 より構成される。

#### 【0025】

定義部 10 では、マルチデータベース統合部 30 で用いる一連の定義情報の登録・修正を行なう。定義部 10 での登録・修正結果は、定義 DB 20 に保存される。定義部は、アクセス権マッピング定義 11 と、定義時アクセス権マッピングの

整合性チェック 1 2 よりなる。アクセス権マッピング定義 1 1 は、マルチデータベース統合部上にて管理する仮想表アクセス権と、統合対象 DBMS 上の DBMS アクセス権とを対応づけ、対応関係を保存する処理を行なう。定義時アクセス権マッピングの整合性チェック 1 2 は、仮想表に対するアクセス権と、実表に対するアクセス権のマッピングにおいて不整合が生じていないかを判定し、不整合の発生しているマッピングを、通常と異なる形態で画面表示することにより、ユーザに通知する。図 1 4 に画面例を示す。

#### 【 0 0 2 6 】

マルチデータベース統合部 3 0 では、定義 DB 2 0 に保存された定義情報を用いて、DBMS1 ( 4 0 )、DBMS2 ( 5 0 ) のデータ統合処理を行なう。マルチデータベース統合部 3 0 は、実行時アクセス権マッピングの整合性チェック 3 1、仮想表アクセス権判定 3 2、仮想表アクセス権変換 3 3 よりなる。実行時アクセス権マッピングの整合性チェック 3 1 では、実行時の環境において、定義したアクセス権マッピングに不整合がないか検出し、不整合を検出した場合には、一定のルールに基づいた修復処理を行なう。仮想表アクセス権判定 3 2 では、AP 側で利用する仮想表に対するアクセス権判定を行なう。仮想表アクセス権変換 3 3 では、仮想表アクセス権を統合対象となる DBMS へのアクセス権に変換する処理を行なう。AP

6 0 からマルチデータベース統合部 3 0 に対して仮想表に対する表操作要求を行なうと、マルチデータベース統合部ではこの仮想表に対する要求を DBMS1 ( 4 0 )、DBMS2 ( 5 0 )、に対する表操作要求へと変換する。AP 6 0 からマルチデータベース統合部 3 0 に対する表操作要求は、データベース操作の標準言語である SQL を使い、マルチデータベース統合部 3 0 から DBMS1 ( 4 0 )、DBMS2 ( 5 0 ) への操作にも SQL を用いる。

#### 【 0 0 2 7 】

DBMS1 ( 4 0 ) および DBMS2 ( 5 0 ) は統合対象となる DBMS である。仮想表に対するアクセス権情報は、仮想表アクセス権 2 4 として定義 DB 上に格納されているが、統合 DBMS 上に格納されている実表に対するアクセス権は、図 1 の DBMS アクセス権 4 1 に示すように、統合対象 DBMS 上に格納されている。DBMS アクセス権 4 1 は、ID 1 0 1、実表名 1 0 2、ユーザ名 1 0 3、表操作権限 1 0 4 よりなる。なお

、統合対象DBアクセス権 4 1 の実装レベルでの詳細は各DBMS毎に異なるが、いずれのDBMSにおいてもDBMSアクセス権 4 1 と同様な項目を内部情報として管理している。そこで本実施例では、説明の簡略化を行なうため、いずれのDBMS上においても統合対象DBアクセス権 4 1 の構造によって、実表に対するアクセス権を管理しているものと仮定して説明を行なう。実際には、DBMSアクセス権 4 1 と異なり形態の表およびファイル等で管理されていてもかまわない。

## 【 0 0 2 8 】

定義DB 2 0 には、マルチデータベース統合部 3 0 を制御するための定義情報を格納する。定義DB 2 0 は、DBMS情報 2 1、ユーザ情報 2 2、仮想表定義 2 3、仮想表アクセス権 2 4、アクセス権マッピング 2 5 を保持する。DBMS情報 2 1 では、統合対象DBMSの管理情報を保持する。ユーザ情報 2 2 では、仮想表ユーザを管理する。仮想表定義 2 3 では、仮想表の定義を格納する。仮想表アクセス権 2 4 では、仮想表に対するアクセス権を管理する。アクセス権マッピング 2 5 では、仮想表アクセス権と、統合対象DBMSのアクセス権との対応関係を保持する。実施例1では、図 4 に詳細を示すアクセス権マッピング 2 5 を用いる。

## 【 0 0 2 9 】

以上で、実施例 1 のブロック図の説明を終える。

## 【 0 0 3 0 】

以下では、図 1 の定義DB 2 0 に格納されている各情報についての詳細説明を、図 3 を用いて行なう。

## 【 0 0 3 1 】

DBMS情報 2 1 は、統合対象となるDBMSを管理するための情報を格納するものであり、図 3 に示すようにDBMS配置情報 3 8 0、DBMSユーザ情報 3 1 0、実表情報 3 2 0、実カラム情報 3 3 0、の 4 つの表よりなる。

## 【 0 0 3 2 】

図 3 のDBMS配置情報 3 8 0 は、DBMS名 3 8 1、DBMSタイプ 3 8 2、ホスト名 3 8 3 よりなる。DBMS名 3 8 1 には、統合対象となるDBMS名を格納する。DBMSタイプ 3 8 2 には、DBMSの種別を格納する。例えばDBMSの製品名などを格納する。ホスト名 3 8 3 には、DBMSが配置されている計算機を表わすホスト名を格



納する。

#### 【 0 0 3 3 】

図 3 の DBMS ユーザ情報 3 1 0 は、DBMS\_UID 3 1 1、DBMS 名 3 1 2、DB 接続ユーザ名 3 1 3、パスワード 3 1 4 よりなる。DBMS\_UID 3 1 1 には、DBMS ユーザ情報 3 1 0 のレコードを一意に識別する識別値を格納する。DBMS 名 3 1 2 には、DBMS 名 3 8 1 に対応する値を格納する。DBMS 接続ユーザ名 3 1 3、パスワード 3 1 4 には、統合対象となる DBMS にたいしアクセスを行なうためのユーザ名およびパスワードを格納する。

#### 【 0 0 3 4 】

図 3 の実表情報 3 2 0 は、RTBL\_ID 3 2 1、実表名 3 2 2、DBMS 名 3 2 3 よりなる。RTBL\_ID 3 2 1 には、実表情報 3 2 0 のレコードを一意に識別する識別値を格納する。実表名 3 2 2 には、DBMS 上に存在する実表の名称を格納する。DBMS 名 3 2 3 には、DBMS 配置情報 3 8 0 の DBMS 名 3 8 2 に格納されている値のうち、対応する値を格納する。

#### 【 0 0 3 5 】

図 3 の実カラム情報 3 3 0 は、RC\_ID 3 3 1、実カラム名 3 3 2、データ型 3 3 3、RTBL\_ID 3 3 4、よりなる。RC\_ID 3 3 1 には、実カラム情報 3 3 0 のレコードを一意に識別する識別値を格納する。実カラム名 3 3 2 には、実表のカラム名称を格納する。データ型 3 3 3 には、実カラム名 3 3 2 の実表におけるデータ型を格納する。RTBL\_ID 3 3 4 には、実カラム名 3 3 3 を保持する実表名に対応した、実表情報 3 2 0 上の RTBL\_ID 3 2 1 の値を格納する。

#### 【 0 0 3 6 】

図 1 の仮想表定義 2 2 は、図 3 に示すように仮想表ー表定義 3 4 0、仮想表ーカラム定義 3 5 0、の 2 つの表よりなる。

#### 【 0 0 3 7 】

図 3 の仮想表ー表定義情報 3 4 0 には、仮想表を定義するための情報を格納する。仮想表ー表定義情報 3 4 0 は、仮想表名 3 4 1、仮想表名定義 SQL 3 4 2 よりなる。仮想表名 3 4 1 には、仮想表名を格納する。仮想表名定義 SQL 3 4 2 には、仮想表を定義するための SQL 文を格納する。この SQL 文は統合対象となる DBMS

上の表を対象として記述されるものである。なお、仮想表を定義する方法は、SQLを用いる手段に限らず、他の手段を用いてもよい。

## 【 0 0 3 8 】

図 3 の仮想表－カラム定義情報 3 5 0 には、仮想表上の項目を管理する情報を格納する。仮想表－カラム定義情報 3 5 0 は、VC\_ID 3 5 1、仮想表名 3 5 2、仮想表カラム名 3 5 3、RC\_ID 3 5 4 よりなる。VC\_ID 3 5 1 には、仮想表項目情報 3 5 0 のレコードを一意に識別する識別値を格納する。仮想表名 3 5 2 には、仮想表名 3 4 1 の値のうち対応する値を格納する。仮想表カラム名 3 5 3 には、仮想表のカラム名称を格納する。RC\_ID 3 5 4 には、仮想表カラムと対応する実カラムの識別子として、実カラム情報 3 3 0 における RC\_ID 3 3 1 の値を格納する。

## 【 0 0 3 9 】

図 1 のユーザ情報 2 3 は、仮想表に対するユーザ名、パスワードを管理するものであり、図 3 のユーザ情報 2 3 に示すように、ユーザ情報 2 3 のレコードを一意に識別する UID 3 7 1 と、ユーザ名 3 7 2、パスワード 3 7 3 よりなる。

## 【 0 0 4 0 】

図 1 の仮想表アクセス権 2 4 は、図 3 に示すように VA\_ID 3 6 1、仮想表名 3 6 2、ユーザ名 3 6 3、仮想表操作権限 3 6 4 よりなる。VA\_ID 3 6 1 には、仮想表アクセス権 2 4 のレコードを一意に識別する識別値を格納する。仮想表名 3 6 2 には、アクセス権対応の対象となる仮想表名を格納する。この仮想表名は仮想表－表定義 3 4 0 の仮想表名 3 4 1 に格納されている仮想表名のうち、該当するものを格納する。ユーザ名 3 6 3 には、ユーザ情報 2 3 に格納されているユーザ名 3 7 2 のうち、仮想表に対するアクセス権を対応づけるものを格納する。仮想表操作権限 3 6 4 には、仮想表表に対する SQL 操作種別の SELECT、DELETE、UPDATE、INSERT のいずれが許可されているかを示す値を格納する。図 3 の例では、仮想表 VT1 に対してユーザ APUser1 が、SELECT と UPDATE の操作が許可されていることを示している。

## 【 0 0 4 1 】

アクセス権マッピング情報 2 5 は、すでに説明したように、仮想表に対して定

義したアクセス権と、統合対象となるDBMS上で定義されているアクセス権を対応づけるものである。実施例1におけるアクセス権マッピング情報25を図4に示す。アクセス権マッピング情報25は、M\_ID401、仮想表名402、DBMS名403、DBMS\_UID404よりなる。M\_ID401には、アクセス権マッピング情報25のレコードを一意に識別する値を格納する。仮想表名402には、図3の仮想表名341のうち対応する値を格納する。DBMS名403には、対応するDBMS名381の値を格納する。DBMS\_UID404には、対応するDBMS\_UIDの値を格納する。DBMS\_UIDからは、DBMSユーザ情報310を参照することにより、DBMSユーザ名を取得できるため、結果的には、DBMSのこのアクセス権マッピング情報25を用いることにより、仮想表と統合対象DBMSのユーザ名とを対応づける。統合対象DBMS上では、ユーザ名を含むアクセス権情報が管理されているため、結果的にアクセス権マッピング情報25は、仮想表アクセス権24とDBMSアクセス権41, 51とを対応づける。図1、図4にこの関連を示す。

#### 【0042】

なお、本発明での定義DB20上の表スキーマについては、図3、図4に示したものに限らない。図3、図4と同等の情報を管理するのであれば、異なる形態で表スキーマの定義を行なってもかまわない。またDBMSを用いず、独自設計フォーマットのファイルシステムの形態で、定義DB20を実現してもよい。

#### 【0043】

上記の実施例では、図3に示した仮想表アクセス権24によって表単位にアクセス権を設定する場合を示したが、さらに、図3に示すように、仮想表カラムアクセス権26によって仮想表の中のカラム単位でアクセス権を設定できる。仮想表カラムアクセス権26は、仮想表を特定するVAC\_ID381と仮想表名382、アクセス権を設定する仮想表カラム名383、アクセス権限を与えられるユーザ名384、及びアクセス権の内容を示す仮想表操作権限385からなる。仮想表カラムアクセス権26を設けることにより、上記の実施例のように複数のDBMSを表単位で統合した仮想表の中のカラム単位でアクセス権を設定できるが、さらに、複数のDBMSをカラム単位で統合した仮想表の場合、図3に示した仮想表カラムアクセス権26によってカラム単位にアクセス権を設定する必要がある。

## 【 0 0 4 4 】

なお、仮想表アクセス権 2 4 と仮想表カラムアクセス権 2 6 の両方を同時に用いることもできる。同一の仮想表に対して、仮想表アクセス権 2 4 と仮想表カラムアクセス権 2 6 の両方にアクセス権が設定されている場合には、表単位のアクセス権が設定されているものとしてアクセス権判定処理を行なうものとする。

## 【 0 0 4 5 】

図 2 に実施例 1 のシステム構成図を示す。本実施例のシステムは、クライアント PC 1 ( 2 1 0 )、クライアント PC 2 ( 2 2 0 )、定義ツール PC 2 3 0、マルチデータベース統合サーバ 2 4 0、DBMSサーバ 1 ( 2 5 0 )、DBMSサーバ 2 ( 2 6 0 )、広域ネットワーク 2 8 0、ネットワーク 2 9 0、よりなる。クライアント PC 1 ( 2 1 0 )、クライアント PC 2 ( 2 2 0 )、定義ツール PC 2 3 0、マルチデータベース統合サーバ 2 4 0、DBMSサーバ 1 ( 2 5 0 )、DBMSサーバ 2 ( 2 6 0 ) は計算機装置であり、これらは通常の計算機装置が備える CPU、メモリ、記憶装置、キーボード、マウスなどを備えているものとする。図 2 に示すように各処理部を配置する。なお処理部の配置は図 2 に示すものに限らない。これらの処理をすべて一台の計算機装置上で処理してもかまわない。また、図 2 では 2 種類の DBMS しか図示していないが、これについては 1 つ以上の任意の個数を対象とすることができる。ネットワークの構成についても図 4 に示したものに限らず、自由に構成できる。

## 【 0 0 4 6 】

以下では、図 7 から図 1 3 のフロー図を用いて実施例 1 の図 1 における処理ブロックの動作について説明する。

## 【 0 0 4 7 】

図 7 を用いて定義部 1 0 の処理フローを説明する。

## 【 0 0 4 8 】

ステップ 7 0 1 では、入力画面を用いて DBMS 情報 2 1 の定義を行う。入力画面の構成は、図 3 の DBMS 情報 2 1 における各テーブルをそのまま表示したものを用いる。

## 【 0 0 4 9 】

ステップ 7 0 2 では、入力画面を用いて、ユーザ情報 2 2 の定義を行う。入力画面の構成は、図 3 のユーザ情報 2 2 の各テーブルをそのまま表示したものを用いる。

## 【 0 0 5 0 】

ステップ 7 0 3 では、入力画面を用いて、仮想表定義 2 3 の定義を行う。入力画面の構成は、図 3 の仮想表定義 2 3 の各テーブルをそのまま表示したものを用いる。

## 【 0 0 5 1 】

ステップ 7 0 4 では、入力画面を用いて、仮想表アクセス権 2 4 の定義を行う。入力画面の構成は、図 3 の仮想表アクセス権 2 4 のテーブルをそのまま表示したものを用いる。

## 【 0 0 5 2 】

ステップ 7 0 5 では、図 1 のアクセス権マッピング定義 1 1 を行う。本処理により定義 DB 2 0 上のアクセス権マッピング 2 5 を定義する。本ステップにおいては、図 1 4 に示す入力画面を用いる。詳細は図 9 を用いて説明する。

## 【 0 0 5 3 】

ステップ 7 0 6 では、図 1 の定義時アクセス権マッピングの整合性チェック 1 2 を行う。

## 【 0 0 5 4 】

本処理により、アクセス権マッピング 2 5 と、これに関連する情報との間で、不整合が生じているか否かのチェックを行う。具体的には、「(1) DBMS ユーザ情報 3 1 0 上のユーザが実際の統合対象側 DBMS 上に存在するか」、「(2) 仮想表操作権限と実表操作権限の間に矛盾がないか」について調べる。詳細は図 1 0 を用いて説明する。

## 【 0 0 5 5 】

なお定義部 1 0 で用いる入力画面は上記ステップで述べたテーブル構成をそのまま表示する画面に限定せず、各ステップで入力対象となる項目を表示するものであれば、どのような画面構成であってもかまわない。更に、図 7 の定義部 1 0

の処理フローは、説明を簡易化するために、定義DBに必要な情報をステップ毎に入力する形で記述されているが、図7の処理フローに限らず、例えばメイン画面のメニューに入力対象となる定義DB上のテーブル名一覧を表示し、このメニューを選択することにより、定義DB上の各テーブルに入力を行う個別画面を表示し、個別テーブルへの入力処理を終了するとメイン画面に戻る、といった、イベントドリブンな処理フロー構成であってもかまわない。

#### 【0056】

図9を用いてアクセス権マッピング定義11の処理フローを説明する。

#### 【0057】

ステップ901では、仮想表ー表定義340とDBMSユーザ情報310を参照することにより、仮想表名および統合対象DBMS側ユーザ名の一覧を取得し、これらを入力画面上に表示する。図14にGUIの一例を示す。このGUIでは仮想表名の一覧を縦軸方向に表示し、DBMS側ユーザ名の一覧を横軸方向に表示している。

#### 【0058】

ステップ902では、仮想表名と統合対象DBMS間の対応づけを自動入力する。図14の例では、ユーザがメニュー項目「自動マッピング1402」をマウスなどにより選択すると、仮想表名とユーザ名の間で名称文字列の包含関係が成立するか否かを判定し、包含関係が成立する場合には、対応個所のセル部分に丸印を表示する。図14では、仮想表名DB1\_User1\_VT3の中に、ユーザ名であるDB1\_User1が文字列として含まれているため、セル1403に示すように自動マッピング結果である丸印が表示される。一般に仮想表名とユーザ名の間に明示的な関係はないが、仮想表定義を行なう際に、ある特定のルールに従って仮想表の名称を設計することが想定できる。こうした特定のルールに従って仮想表名を定義している場合、本ステップの自動マッピング処理により、手入力でマッピング定義する作業を軽減する。以下にルールの概要を説明する。なお自動マッピングのルールは、ここで述べたルールに限定されない。

#### 【0059】

ステップ903では、仮想表名と統合対象DBMS間の対応づけを手動で入力する。図XXの例では、対応づけたい個所のセル部分をマウスでクリックすることによ

り、セルに丸が表示される。ステップ 9 0 2 およびステップ 9 0 3 において、図 1 4 の画面上のセルに対して丸を表示することにより、仮想表名と統合対象 DBMS 間の対応づけを行う。

#### 【 0 0 6 0 】

ステップ 9 0 4 では、ユーザがメニュー項目「登録 1 4 0 1」をマウスなどにより選択することにより、アクセス権マッピング情報 2 5 の格納処理を行なう。ステップ 9 0 2、9 0 3 においてセルに丸が表示された部分のマッピング関係を、アクセス権マッピング情報 2 5 として格納する。まず画面上にて丸を指定されているセルに対応した仮想表名と DBMS ユーザ名を取得する。画面から取得した DBMS ユーザ名を検索条件として、DBMS ユーザ情報 3 1 0 を参照することにより、DBMS\_UID の値を得ることができるので、これらを組みにしてアクセス権マッピング情報 2 5 に格納する。

#### 【 0 0 6 1 】

図 1 0 を用いて定義時アクセス権マッピングの整合性チェック 1 2 の処理フローを説明する。

#### 【 0 0 6 2 】

ステップ 1 0 0 1 では、統合対象 DBMS 側のユーザ名確認を行う。これは DBMS ユーザ情報 3 1 0 に格納されているユーザ名、パスワードを用いて各 DBMS への接続処理を実際に行うことにより確認する。接続できないユーザ名、パスワードが存在した場合には、これに対応する DBMS\_UID をメモリ上に保持する。

#### 【 0 0 6 3 】

ステップ 1 0 0 2 では、仮想表定義 2 2 および仮想表アクセス権 2 4 から、アクセス権マッピング 2 5 の整合性確認に用いる整合性条件を導出する。以下では整合性条件の導出方法を説明する。整合性条件を導出するには、図 3 の仮想表一表定義 3 4 0 および仮想表アクセス権 2 4 から、各実表単位に必要な操作権限を導出する。例えば仮想表 VT1 が「SELECT RT1\_1.C1, RT1\_1.C2, RT2\_1.C3, RT2\_1.C4 FROM RT1\_1, RT2\_1 WHERE RT1\_1.C1= RT2\_1.C1」といった SQL によって定義されているものとする。ここで、RT1\_1、RT2\_1 はおのおの DBMS1 (4 0)、DBMS2 (5 0) 上の実表とする。図 3 の仮想表アクセス権 2 4 に示すように、仮想表 VT1

に対し、SELECT、UPDATE権限が与えられている。仮想表VT1に対してSELECT、UPDATE権限を与えるには、VT1を実際に構成する実表RT1\_1、RT2\_1に対してSELECT、UPDATE権限が必要なことがわかる。

#### 【 0 0 6 4 】

次にアクセス権マッピング情報 2 5 を参照することにより、VT 1 においては、DBMS1のユーザとしてDB1\_User1が、DBMS2のユーザとしてDB1\_User2が対応づけられていることがわかる。

#### 【 0 0 6 5 】

この対応づけに基づいて、仮想表VT1においては、定義DB 2 0 と、各統合DBMS上のアクセス権定義情報（DBMSアクセス権 4 1、DBMS 2 アクセス権 5 1）との間で整合性が成立するためには、以下の条件が成立することが必要であることが導出できる。

#### 【 0 0 6 6 】

（整合性条件 1）：DB1\_User 1 がRT1\_1に対してSELECT権限とUPDATE権限をもつ

。

（整合性条件 2）：DB2\_User 1 がRT2\_1に対してSELECT権限とUPDATE権限をもつ

。

ここでのべた整合性条件の導出手順を、ステップ 1 0 0 2 での処理として行なう

。

#### 【 0 0 6 7 】

ステップ 1 0 0 3 では、統合対象DBMS側からDBMSアクセス権 4 1、5 1 を取得する。各DBMS毎に定義されたユーザ名、パスワードを用いて、統合対象DBMS側に接続を行なう。そして各ユーザ対応したアクセス権をDBMS1アクセス権 4 1、DBMS2アクセス権 5 1 から取得する。なお本ステップの代案として、DBMSユーザ情報 3 1 0 上において、DBMS単位にスーパーユーザ（本実施例で必要となるDBMSの管理情報を全て参照できる特権ユーザ）を設けることで、統合対象DBMSからDBMSアクセス権 4 1、5 1 を取得する実現方式を用いることもできる。

#### 【 0 0 6 8 】

ステップ 1 0 0 4 では、整合性成立条件と統合対象DBMS上から取得したアクセ



ス権を付き合わせるにより、整合性が成立しているか否かの判定を行う。例として、DBMS 1 上のDB1\_User1に対しては、RT1\_1へのSELECT、UPDATE権限が与えられおり、DBMS2上のDB2\_User1に対しては、RT2\_1へのSELECT権限のみが与えられているものとする。こうした場合、DBMS2上のDB2\_User1に対しては、RT2\_1へのUPDATE権限が与えられていないため、ステップ 1 0 0 2 の整合性条件 2 が成立しない。よって不整合が生じているものと判定する。本ステップでは、ステップ 1 0 0 2 の整合性条件が全て成立した時にのみ、整合性が成立しているものと判定する。なおステップ 1 0 0 1 においてDBMS上に存在しないと判定されたユーザー名を整合性条件が含んでいる場合には、その時点で不整合が生じているものと判定する。

#### 【 0 0 6 9 】

ステップ 1 0 0 5 では、不整合が生じている修正対象個所を、修正対象外の個所と区別できる形態で画面表示する。例えば、図 1 4 のセル 1 4 0 4 の様に「VT 1とDB2\_User1とのマッピングを与えている部分のセルを強調表示する」などの画面制御を行う。

#### 【 0 0 7 0 】

次に図 8 を用いてマルチデータベース統合部 3 0 の処理フローを説明する。

#### 【 0 0 7 1 】

ステップ 8 0 1 では、定義DB 2 0 の情報をメモリ上に読みこむ処理を行う。

#### 【 0 0 7 2 】

ステップ 8 0 2 では、実行時アクセス権マッピングの整合性チェック 3 1 を行う。本処理では、メモリ上のアクセス権マッピング情報を実行時の環境と比較して不整合が発生していないかの検出を行う。不整合が発生した場合には、メモリ上のアクセス権マッピング情報に対して自動修正を行う。図 1 1 において詳細な説明を行なう。

#### 【 0 0 7 3 】

ステップ 8 0 3 では、APからの接続要求を待機する。APからの接続要求があった場合に、ステップ 8 0 4 を実行する。

## 【 0 0 7 4 】

ステップ 8 0 4 では、ユーザ認証処理を行う。ユーザ認証は、APからの接続要求時のユーザ名とパスワードがユーザ情報 2 3 に存在するか否かにより、ユーザ認証の判定を行なう。ユーザ名とパスワードの双方の値が一致するレコードが存在した場合にのみ、ユーザ認証が成立したと判定する。

## 【 0 0 7 5 】

ステップ 8 0 5 ではユーザ認証成功の判定を行う。成功した場合にはステップ 8 0 6 を行う。失敗した場合にはAP 6 0 とマルチデータベース統合部 3 0 との接続を行わず、ステップ 8 0 3 の状態に戻る。

## 【 0 0 7 6 】

ステップ 8 0 6 では、AP 6 0 とマルチデータベース統合部 3 0 との接続を、従来のDBMSと同様な方法にて確立する。

## 【 0 0 7 7 】

ステップ 8 0 7 では、接続を確立したAP 6 0 からのSQL要求を待機する。SQL要求があった場合には、ステップ 8 0 8 を実行する。

## 【 0 0 7 8 】

ステップ 8 0 8 では、仮想表アクセス権判定 3 2 の処理を行う。ここではマルチデータベース統合部 3 0 への接続ユーザに対して、SQLで要求された仮想表および仮想表への操作権限が認可されるか否かの判定を行う。図 1 2 において詳細な説明を行なう。

## 【 0 0 7 9 】

ステップ 8 0 9 では、仮想表アクセス権の判定に成功したか否かの判定を行う。成功した場合にはステップ 8 1 0 を実行する。失敗した場合には、エラーメッセージの出力などを行ない、ステップ 8 0 7 のSQL要求待ちの状態に遷移する。

## 【 0 0 8 0 】

ステップ 8 1 0 では、仮想表アクセス権変換処理 3 3 を行う。ここでは、SQLで要求されている仮想表名を検索条件とすることで、アクセス権マッピング 2 5 からDBMS\_UID 4 0 4 を取得する。そしてDBMSユーザ情報 3 1 0 から、DBMS\_UID 4 0 4 に対応するDBMS接続ユーザ名 3 1 1、パスワード 3 1 4 を取得する。

## 【 0 0 8 1 】

ステップ 8 1 1 では、上記ステップにより得た統合対象 DBMS 接続ユーザ名、パスワードを用いて、統合対象 DBMS 側とマルチデータベース統合部 3 0 との接続を確立する。そして統合対象の DBMS 単位に SQL を実行し、実行結果をマルチデータベース統合部 3 0 において統合し、AP 6 0 に対して統合結果を戻す。本ステップの処理には、既存技術のマルチデータベース統合システムにおけるデータ統合処理を用いる。

## 【 0 0 8 2 】

図 1 1 を用いて実行時アクセス権マッピングの整合性チェック 3 1 の処理フローを説明する。

## 【 0 0 8 3 】

ステップ 1 1 0 1 では、図 1 0 のステップ 1 0 0 1 と同様な、統合対象 DBMS 側ユーザ名の確認を行う。

## 【 0 0 8 4 】

ステップ 1 1 0 2 では、図 1 0 のステップ 1 0 0 2 と同様な処理を行う。仮想表定義および仮想表アクセス権から、整合性を確認するための整合性条件を導出を行う。

## 【 0 0 8 5 】

ステップ 1 1 0 3 では、図 1 0 のステップ 1 0 0 3 と同様な処理を行う。統合対象 DBMS から実表に対するアクセス権情報 4 1、5 1 を取得する。

## 【 0 0 8 6 】

ステップ 1 1 0 4 では、図 1 0 のステップ 1 0 0 4 と同様な処理を行う。整合性成立条件と統合対象 DBMS 上から取得したアクセス権を付き合わせるにより、整合性が成立しているか否かの判定を行う。図 1 0 で用いた仮想表 VT1 の例を再度用いると、DBMS2 上の DB2\_User1 に対しては、RT2\_1 への UPDATE 権限が与えられていないため、整合成立のため条件が成立せず、不整合が生じているものと判定する。この状態では仮想表 VT1 に対して UPDATE の SQL 実行を行なうことが不可能である。

## 【 0 0 8 7 】

ステップ 1 1 0 5 では、不整合が生じている場合に、本来定義されている接続 DB ユーザ名の代わりに利用できる修復用のユーザ名が、統合対象 DBMS 側に存在するか否の検出を行う。本ステップでは修復ユーザ名を、DBMS ユーザ情報 3 1 0 上の DB 接続ユーザ名 3 1 3 から検出する。仮想表 VT1 の例では、DB2\_User1 以外に、DBMS2 上の RT2\_1 に対し SELECT 権限と UPDATE 権限をもつ別ユーザが存在すれば、VT1 に対して SELECT および UPDATE 処理を行なうことができる。以下ではこうしたユーザのことを修復ユーザ名とよぶ。修復ユーザ名を検出するには、図 1 0 のステップ 1 0 0 3 と同様な処理により、各 DBMS 上の DBMS アクセス権を参照する。そして DBMS 上のユーザが修復ユーザとして必要となる表操作権限をもつかを判定する。この例の場合には、DBMS2 上の RT2\_1 に対し SELECT 権限と UPDATE 権限をもつユーザが DBMS アクセス権 5 1 上に存在するか否かを判定する。以上の処理により、もし存在するのであれば、修復ユーザ名を検出できる。修復ユーザ名を検出した場合には、DBMS ユーザ情報 3 1 0 を参照することにより修復ユーザ名に対応した、DBMS\_UID 3 1 1 を取得する。なお DBMS ユーザ情報 3 1 0 と DBMS アクセス権 4 1、5 1 の設定内容によっては、修復ユーザが検出できない場合もある。

## 【 0 0 8 8 】

ステップ 1 1 0 6 では、ステップ 1 1 0 5 で検出した別ユーザ名に基づいてアクセス権マッピング情報 2 5 の修復を行う。修復ユーザ名を取得できた場合には、不整合が発生しているユーザ名に対応するアクセス権マッピング情報 2 5 の DBMS\_UID 4 0 4 を、修復ユーザ名に対応している DBMS\_UID に変更する。この際、変更を行なったことを、ログ情報などで出力する処理も行なう。

## 【 0 0 8 9 】

ステップ 1 1 0 7 では、修復ユーザ名が存在しなかった場合に、影響を受ける仮想表アクセス権を、仮想表アクセス権 2 4 から検出する。まず、修復不可能な DBMS 接続ユーザ名に対応する DBMS\_UID の値に対応づけられている仮想表名 4 0 2 を、アクセス権マッピング 2 5 より取得する。そして、取得した仮想表名を検索条件とすることにより、仮想表アクセス権 2 4 から、影響を受ける仮想表名に対応した VA\_ID 3 6 1 をメモリ上に保持する。

## 【0090】

図12を用いて仮想表アクセス権判定32の処理フローを説明する。

## 【0091】

ステップ1201では、AP60から問合せされたSQLに含まれている仮想表名、表操作権限種別と、仮想表アクセス権24を比較することにより、仮想表アクセス権が成立するか否かの判定を行う。成立する場合にはステップ1202を実行する。成立しない場合にはステップ1203を実行する。

## 【0092】

ステップ1202では、ステップ1201においてアクセス権成立と判定するために用いた仮想表アクセス権24上のVA\_ID361が、ステップ1107で実行時不整合の影響をうけるものとして検出したVA\_ID361と一致するか否かを判定する。一つでも一致する場合にはステップ1204を実行する。全て一致しない場合にはステップ1205を実行する。

## 【0093】

ステップ1203では、エラーメッセージに「該当アクセス権が定義されていません」を指定し、仮想表アクセス権判定を失敗として戻る。エラーメッセージは、AP60側に画面出力などを行なうことにより通知する。

## 【0094】

ステップ1204では、エラーメッセージに「実行時にアクセス権定義の不整合が発生しました」を指定し、仮想表アクセス権判定を失敗として戻る。エラーメッセージは、AP60側に画面出力などを行なうことにより通知する。

## 【0095】

ステップ1205では、仮想表アクセス権判定を成功として戻る。

## 【0096】

図13を用いて仮想表アクセス権変換33の処理フローを説明する。

## 【0097】

ステップ1301では、図4のアクセス権マッピング25と図3のDBMSユーザ情報310を参照することにより、統合対象DBMSに対するユーザ名、パスワードを取得する。仮想表VT1に対してSQL操作が行なわれる場合には、アクセス権マッ

ピング 2 5 を参照することにより DBMS\_UID の値として 1、2 を得、これに対応するレコード情報を DBMS ユーザ情報 3 1 0 から検索することにより、DB 接続ユーザ名として DB1\_User1 と DB2\_User2 を取得することができる。

【 0 0 9 8 】

以上で実施例 1 の説明を終え、次に実施例 2 について述べる。実施例 2 の処理ブロック図を図 1 5 に示す。以下では実施例 1 との違いについて説明する。実施例 1 との違いを述べる形で、実施例 2 の処理フローを説明する。

【 0 0 9 9 】

実施例 1 と実施例 2 の定義 DB 2 0 が異なる点は、まず実施例 1 のアクセス権マッピング情報 2 5 の代わりに、図 5 に示すアクセス権マッピング情報 5 0 0 を用いることである。図 5 に示すように実施例 2 では、ユーザ名と統合対象 DBMS のユーザ名とを対応づける。

【 0 1 0 0 】

次に、実施例 2 のアクセス権マッピング定義 1 5 0 3 のとアクセス権マッピング定義 1 1 (図 9) は以下の点が異なる。

【 0 1 0 1 】

図 9 におけるステップ 9 0 1 では、仮想表名と統合対象 DBMS 側のユーザ名一覧を画面表示する代わりに、マルチデータベース統合システムにおけるユーザ名一覧と統合対象 DBMS 側のユーザ名一覧を画面表示する。マルチデータベース統合システムにおけるユーザ名一覧は、ユーザ情報 2 2 より取得することができる。ステップ 9 0 2、9 0 3 では、仮想表名と統合対象 DBMS 側のマッピングを定義する代わりに、マルチデータベース統合システムにおけるユーザ名一覧と統合対象 DBMS 側ユーザ名の間のマッピングを定義する。特にステップ 9 0 2 では、マルチデータベース統合システムにおけるユーザ名と統合対象 DBMS 側ユーザ名を比較し、ユーザ名が一致している場合には自動的にマッピング処理を行なう。ステップ 9 0 4 では、実施例 2 におけるアクセス権マッピング情報 5 0 0 に対する格納処理を行う。

【 0 1 0 2 】

実施例 2 の定義時アクセス権マッピングの整合性チェック 1 5 0 4 では、図 1

0のステップ1001における統合対象DBMS側のユーザ名確認のみを行い、他のステップを行わない。

#### 【0103】

実施例2の実行時アクセス権マッピングの整合性チェック1501では、図11のステップ1101、ステップ1107のみ行う。ステップ1101の統合対象DBMS側の確認を行った後には、ステップ1107として確認できなかった統合対象DBMSにより影響をうけるアクセス権マッピング情報500を検出する。

#### 【0104】

実施例2の仮想表アクセス権判定32については、実施例1と同様に行う。

#### 【0105】

実施例2のアクセス権変換1502は、実施例1の仮想表アクセス権変換33で参照するアクセス権マッピング情報25の変わりに、アクセス権マッピング情報500を用いる点が異なる。実施例2では、仮想表名ではなく、ユーザ名に基づいてDBMS\_UIDの取得を行なう。

#### 【0106】

なお、実施例1と実施例2の特殊な場合として、図6に示す方式が考えられる。これは統合対象DBMS単位に固定ユーザ名を対応づける方式であり、統合対象DBMS単位に固定のユーザ名DB1\_User1、DB2\_User1を対応づけている。本実施例は、実施例1および実施例2において、常に固定ユーザ名を対応づけた処理であるため、詳細説明は省略する。

#### 【0107】

図16はアクセス権を対応づけるための画面例である。この画面例は、図14にてすでに説明した画面と同等の定義参照を行なうものである。仮想表、およびDB側のユーザ名がアイコン状態で各々左右に表示される。各アイコンをダブルクリックすると、詳細な内容が表示される(1602, 1607)仮想表とユーザ名のアイコン同士をマウスなどで対応づけることでアイコン間にラインが引かれ、このことがリンク関係を示す。エラーチェックを行うと、エラーが発生したライン上が他のラインと異なる形態で表示される。図16では×印が表示されている(1605)。図16に示した以外にラインの太さを変更したり、色を変更し

たりすることによって、エラー発生ラインを表示する様にしてもよい。

#### 【0108】

図17はDBMS自身と他DBMSとの統合を行う統合DBMS1710による実施の形態を示す。本実施形態では、仮想表アクセス権の制御を既に存在しているDBMS上の拡張機能として実施する。本実施形態では、統合DBMS1710のテーブルと外部DBMS140などのテーブルを統合したものを仮想表として定義する。そして、すでに述べた実施例と同様に、統合DBMS1710のDBMSアクセス権1741と、外部DBMS1(40)などのDBMSアクセス権41を用いたアクセス権マッピング定義11を行う。上記の処理で定義した定義情報1730を用いることにより、統合DBMS1710のテーブルと外部DBMS1(40)などのテーブルを統合した仮想表に対して、仮想表アクセス権判定32、仮想表アクセス権変換33などの一連の処理を統合DBMS1710で行う。なお、定義情報1730に格納される情報は、すでに説明した定義DB20と同様であるので詳細な説明は省略する。なお、図17にはアクセス権マッピング25が示されているが、すでに説明した他の実施形態でのアクセス権マッピングの対応づけを、統合統合DBMS1710にて行うことも可能である。

#### 【0109】

図18は、定義ツールが複数のDB統合システムを管理する場合のシステム構成図である。図18は、定義DBの内容が各サブシステムの定義情報を統合管理する。即ち、図18は、統合リポジトリシステムにおける実施の形態を示す。ここで、統合リポジトリシステムとは、DBMS、文書管理、EDIなどのミドルウェアを連携するためのシステム間にまたがるアクセス権やデータ配置などの設定情報(メタデータと呼ばれる)を管理するシステムである。

#### 【0110】

本実施例では、仮想表アクセス権の定義部10を、統合リポジトリシステム1810の一機能として実現する。統合リポジトリシステム1810にて定義した設定情報(メタデータ)を定義DB20から、連携対象となるミドルウェアへ配布する。図18では、定義DB20から、統合DBMS1710a, bなどにメタデータを配布する場合を示す。そして、統合DBMS1710a, bでは、配



布されたメタデータにもとづいて、仮想表アクセス権判定 3 2、仮想表アクセス権変換 3 3 などの一連の処理をそれぞれの統合 DBMS 1 7 1 0 a, b で行う。

【 0 1 1 1 】

【発明の効果】

本発明により、AP側の仮想表に対するアクセス権定義と、統合対象DBMS側のアクセス権定義を並行してシステム構築処理を行なうことができる。また、本方式により、統合対象側のDBMS側において、アクセス対象となる仮想表や、仮想表にアクセスするユーザを切り分けるためのログ情報を、統合対象側DBMSにて取得するために必要なユーザ設定を行なうことが容易になる。

【図面の簡単な説明】

【図 1】

実施例 1 の処理概要図である。

【図 2】

実施例 1 のシステム構成図である。

【図 3】

実施例 1 のテーブル構成図である。

【図 4】

仮想表を単位としたアクセス権マッピングを示す図である。

【図 5】

ユーザを単位としたアクセス権マッピングを示す図である。

【図 6】

DBMSを単位としたアクセス権マッピングを示す図である。

【図 7】

実行部の処理フロー図である。

【図 8】

マルチデータベース統合部の処理フロー図である。

【図 9】

アクセス権マッピング定義の処理フロー図である。

【図 1 0】

定義時アクセス権マッピングの整合性チェックのフロー図である。

【図 1 1】

実行時アクセス権マッピングの整合性チェックのフロー図である。

【図 1 2】

仮想表アクセス権判定のフロー図である。

【図 1 3】

アクセス権変換のフロー図である。

【図 1 4】

アクセス権マッピング定義画面例である。

【図 1 5】

実施例2の処理概要図である。

【図 1 6】

アクセス権を対応付けるための画面例である。

【図 1 7】

DBMS自身と他のDBMSとの統合を行う場合のシステム構成図である。

【図 1 8】

定義ツールが複数のDB統合システムを管理する場合のシステム構成図である。

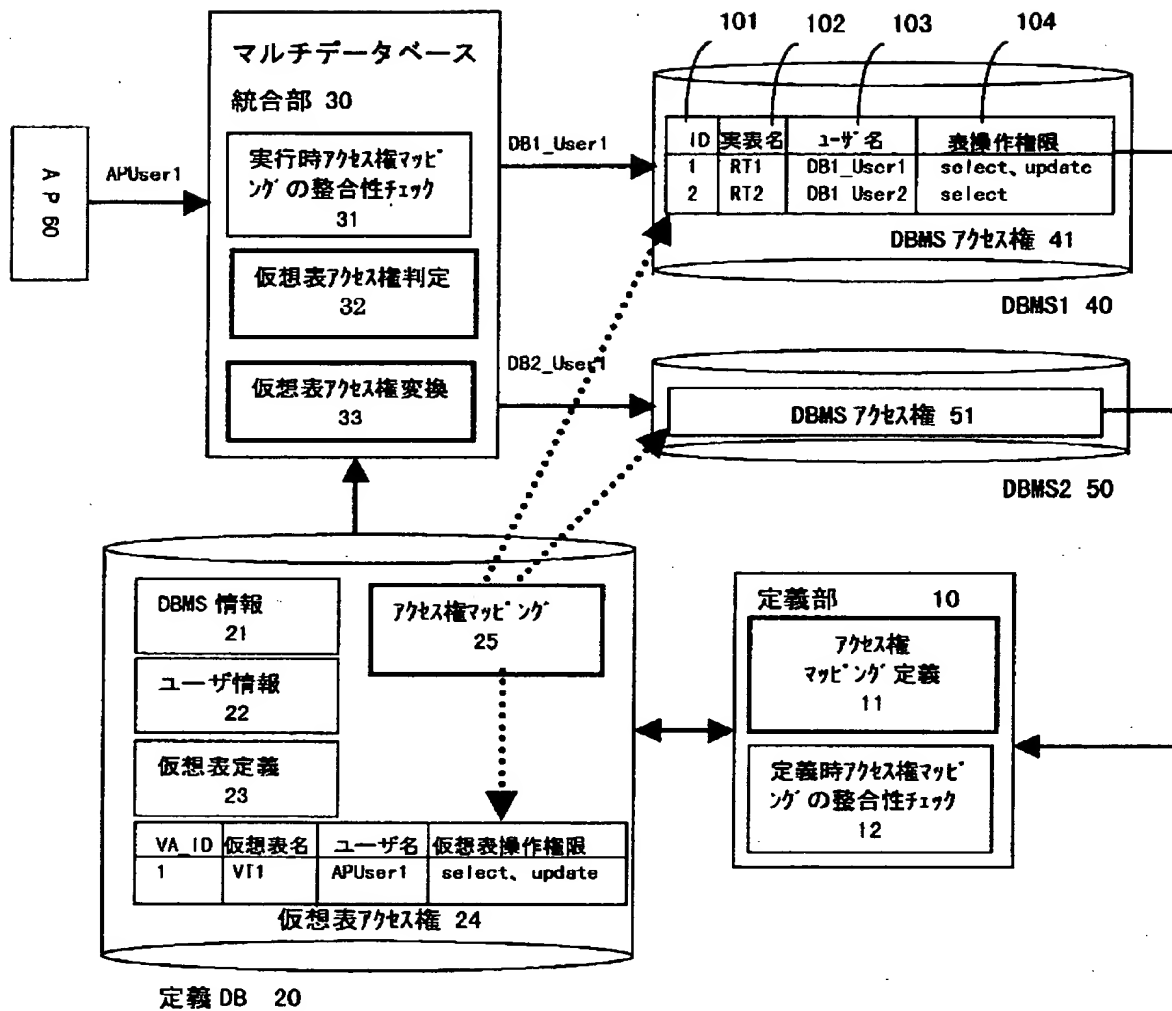
【符号の説明】

- 1 0 . . . 定義部
- 2 0 . . . 定義DB
- 3 0 . . . マルチデータベース統合部
- 4 0 . . . 統合対象DBMS 1
- 5 0 . . . 統合対象DBMS 2
- 6 0 . . . アプリケーション

【書類名】 図面

【図 1】

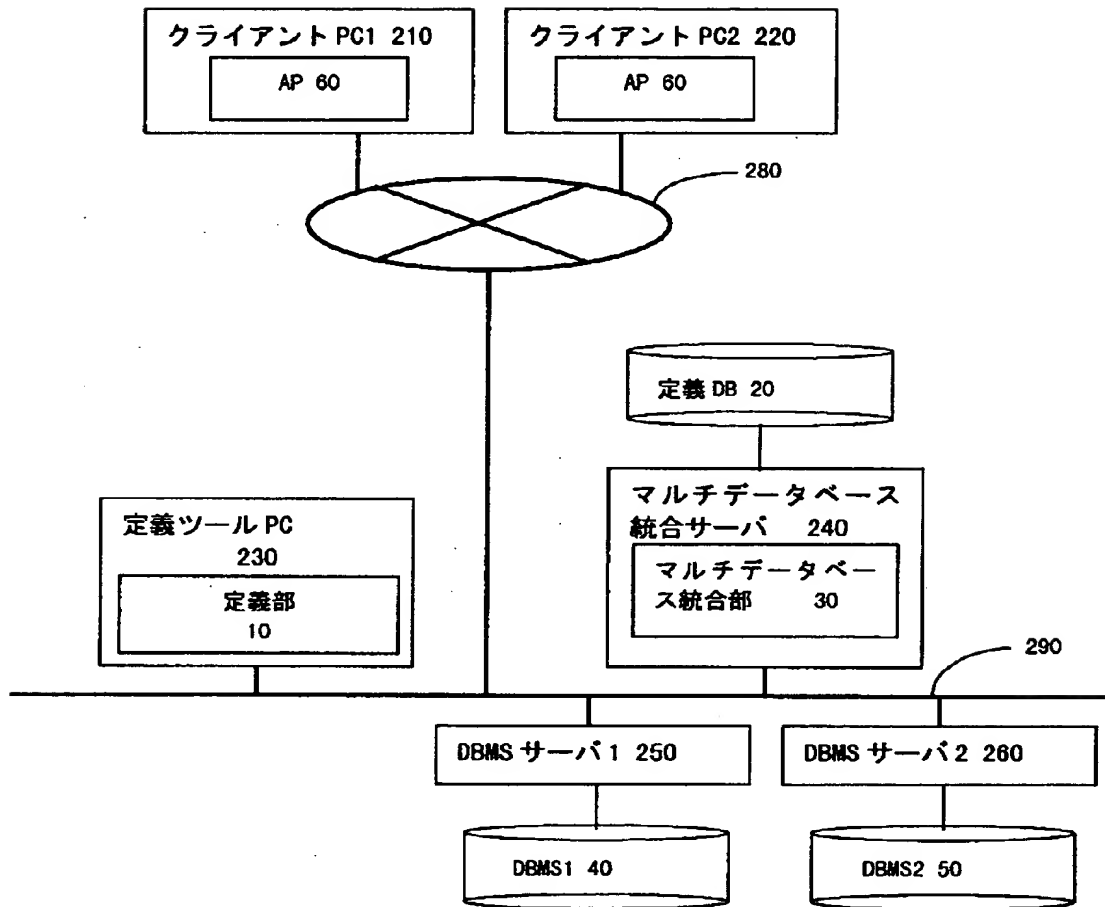
図 1



...▶; データ間の対応 →▶; データの流れ

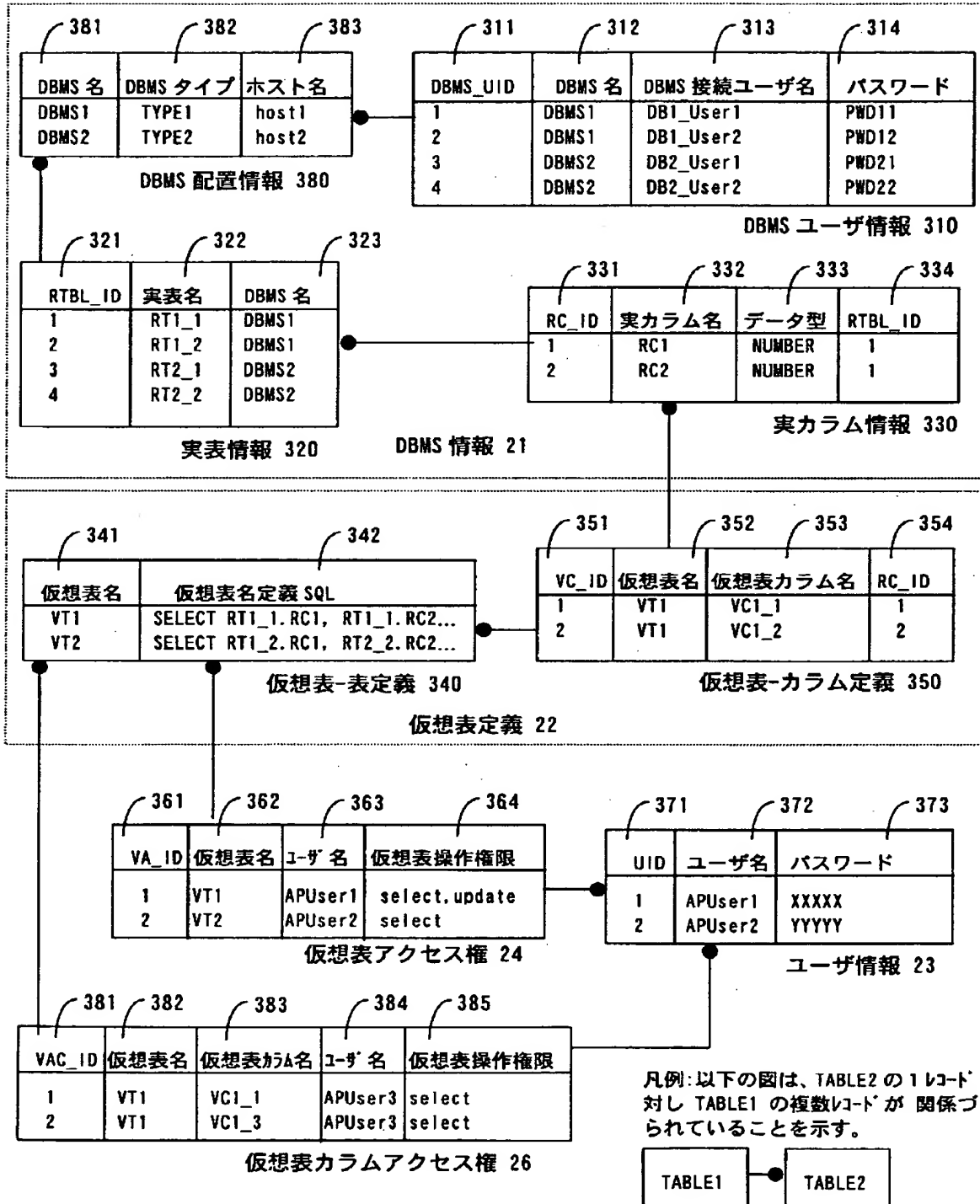
【図 2】

図 2



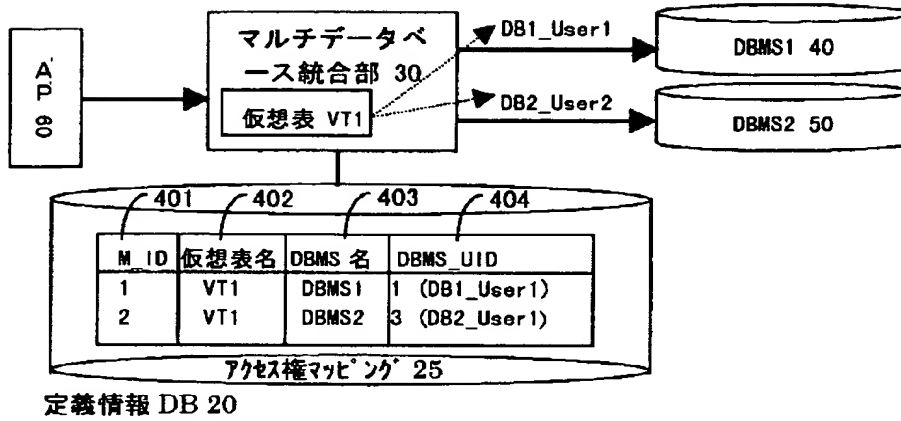
【図 3】

図 3



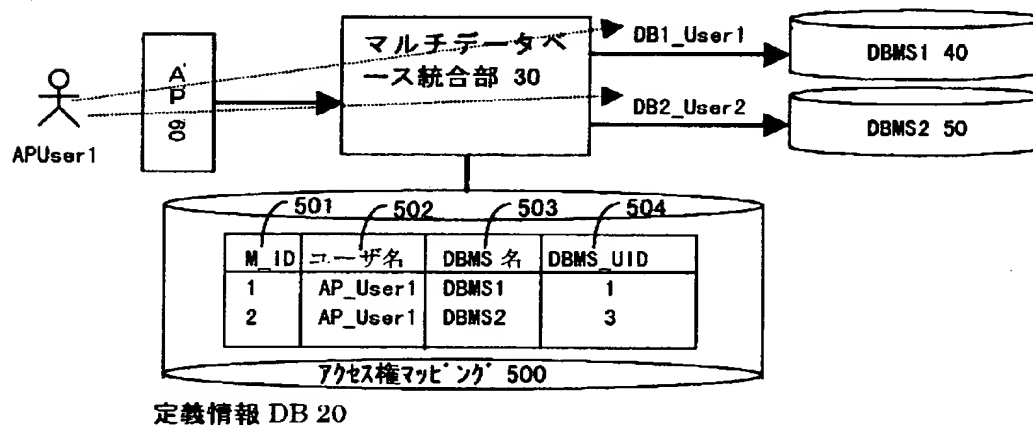
【図 4】

図 4

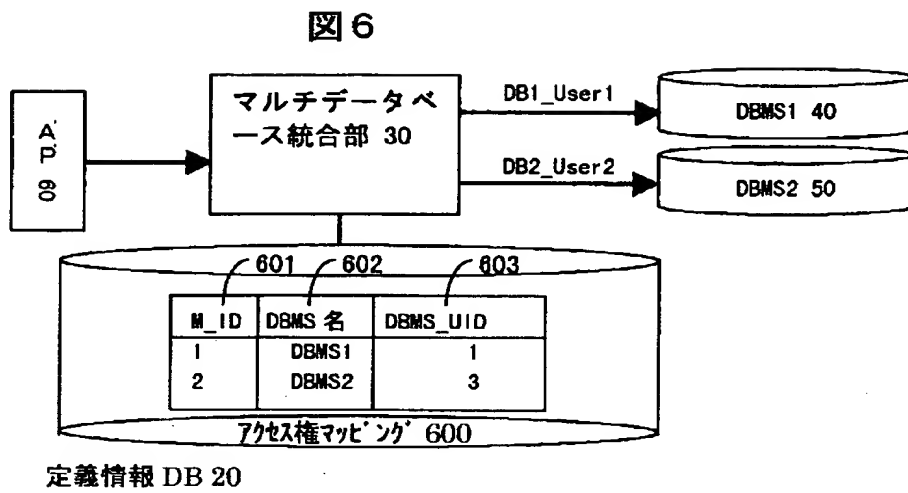


【図 5】

図 5

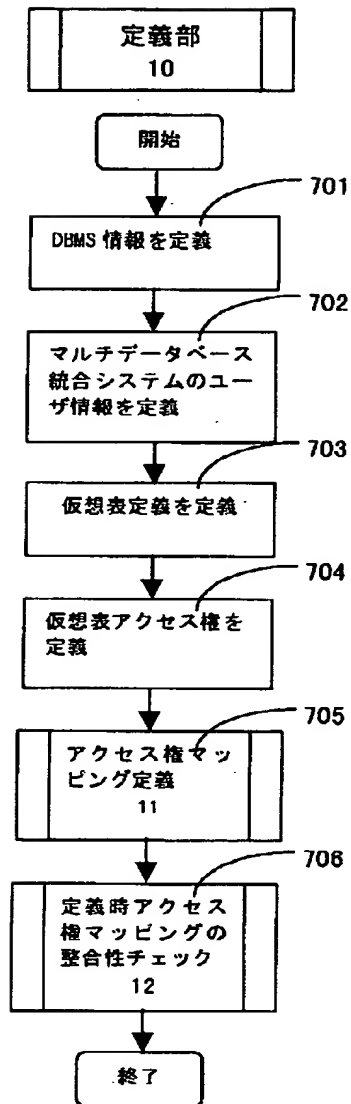


【図 6】



【図 7】

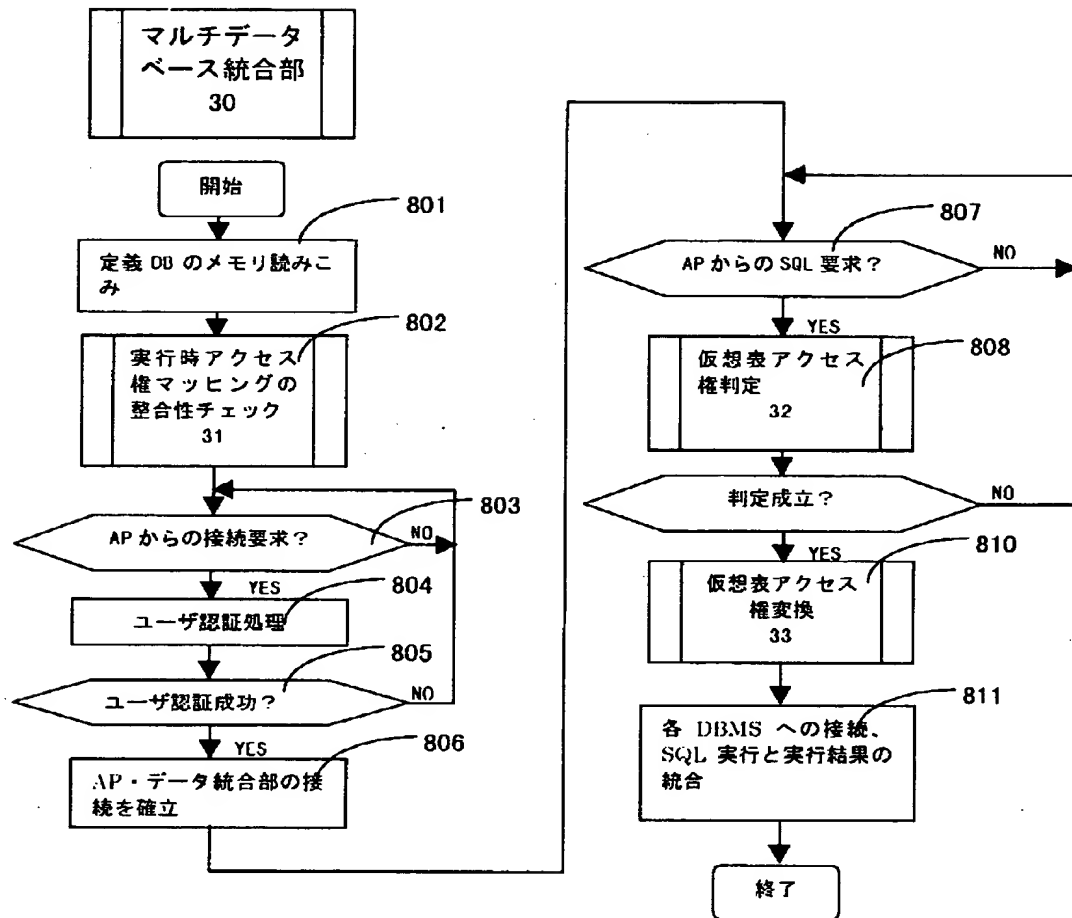
図 7



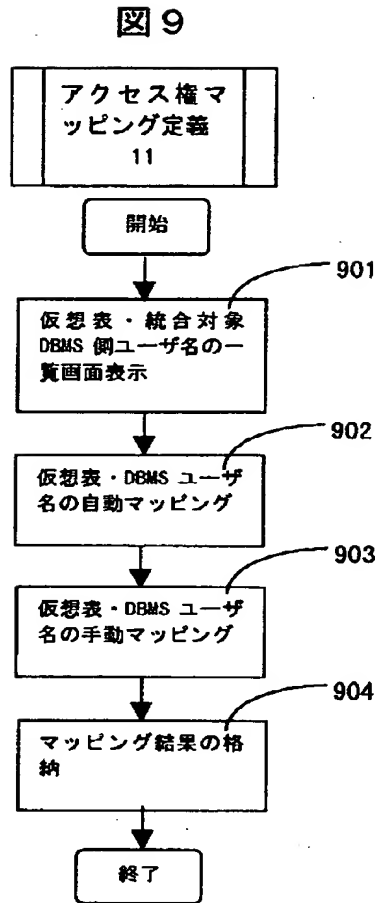


【図 8】

図 8

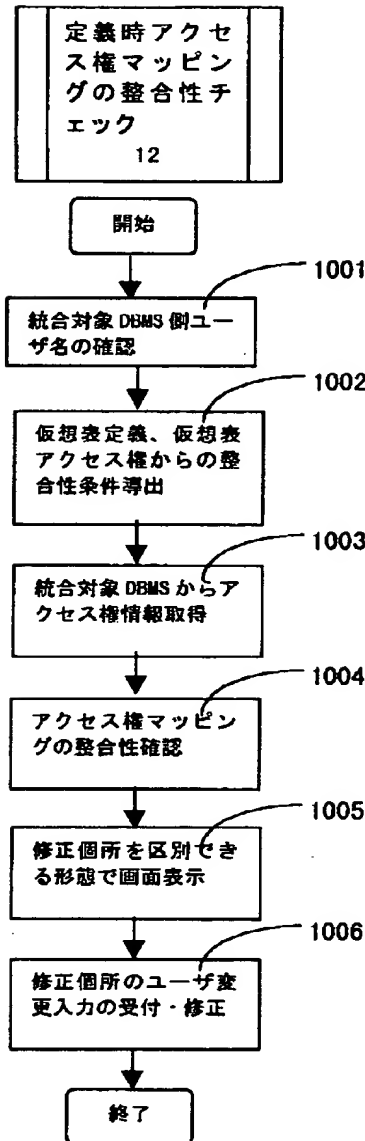


【図 9】



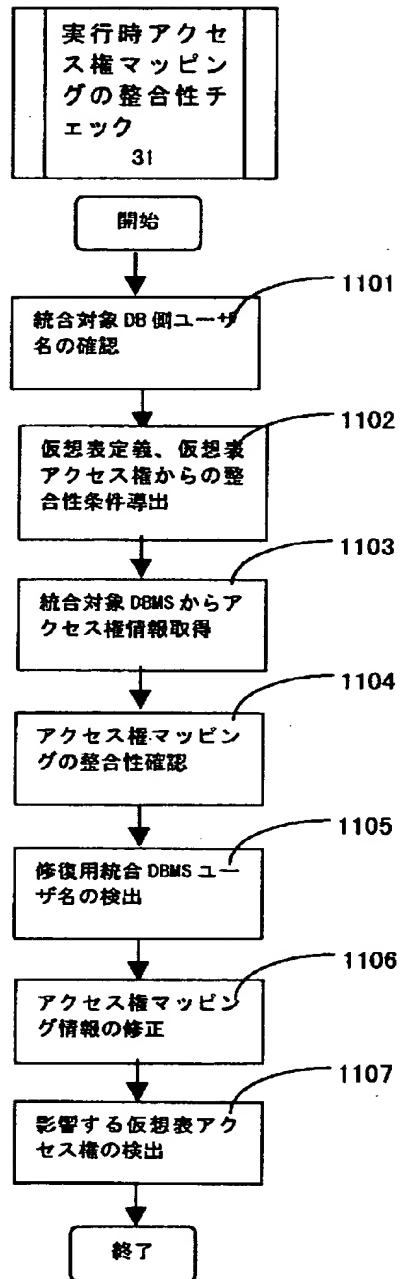
【図 1 0】

図 1 0



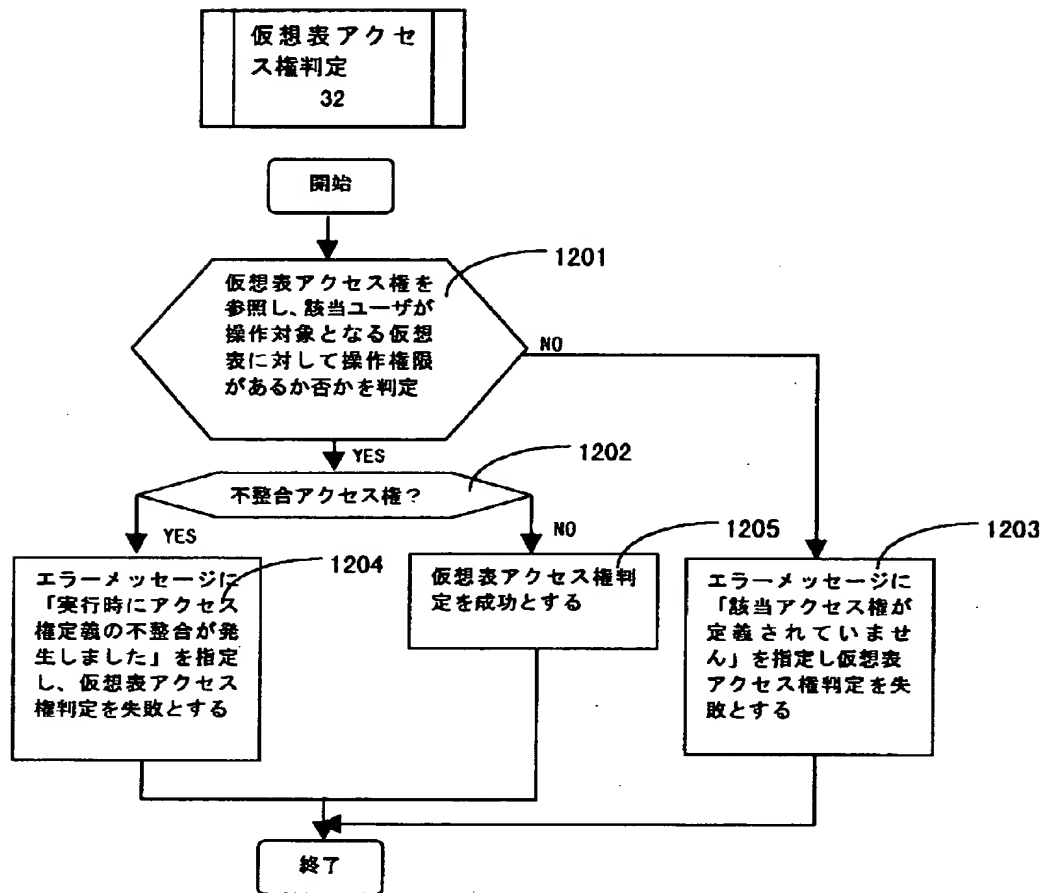
【図 11】

図 11



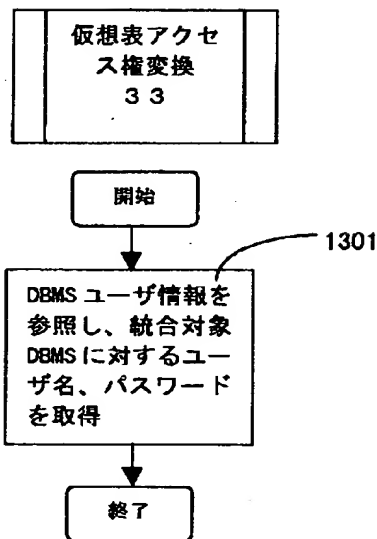
【図 12】

図 12



【図 1 3】

図 1 3



【図 1 4】

図 1 4

登録 自動マッピング					
	DBMS 名	DBMS1		DBMS2	
	ユーザ名	DB1_USER1	DB1_USER2	DB2_USER1	DB2_USER2
仮想表名					
VT1		○		○	
VT2			○		○
DB1_User1_VT3		○			

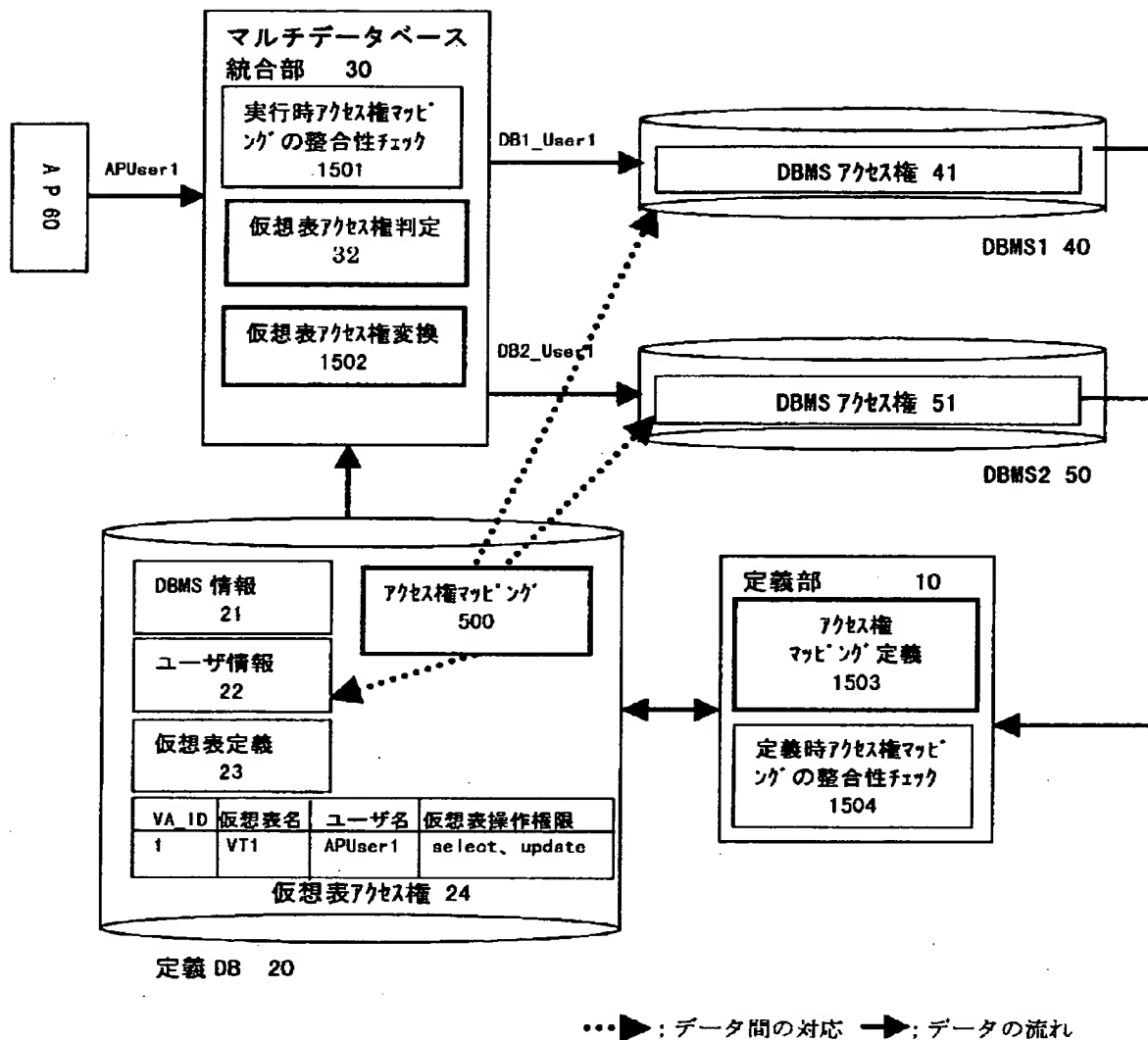
図 14 は、登録 自動マッピングの表を示しています。表の構造は以下の通りです。

	DBMS 名	DBMS1		DBMS2	
	ユーザ名	DB1_USER1	DB1_USER2	DB2_USER1	DB2_USER2
仮想表名					
VT1		○		○	
VT2			○		○
DB1_User1_VT3		○			

注釈: 1401 は「登録 自動マッピング」のタイトル、1402 は「仮想表名」の列、1403 は「DB1\_USER1」の列、1404 は「DB2\_USER1」の列を指します。

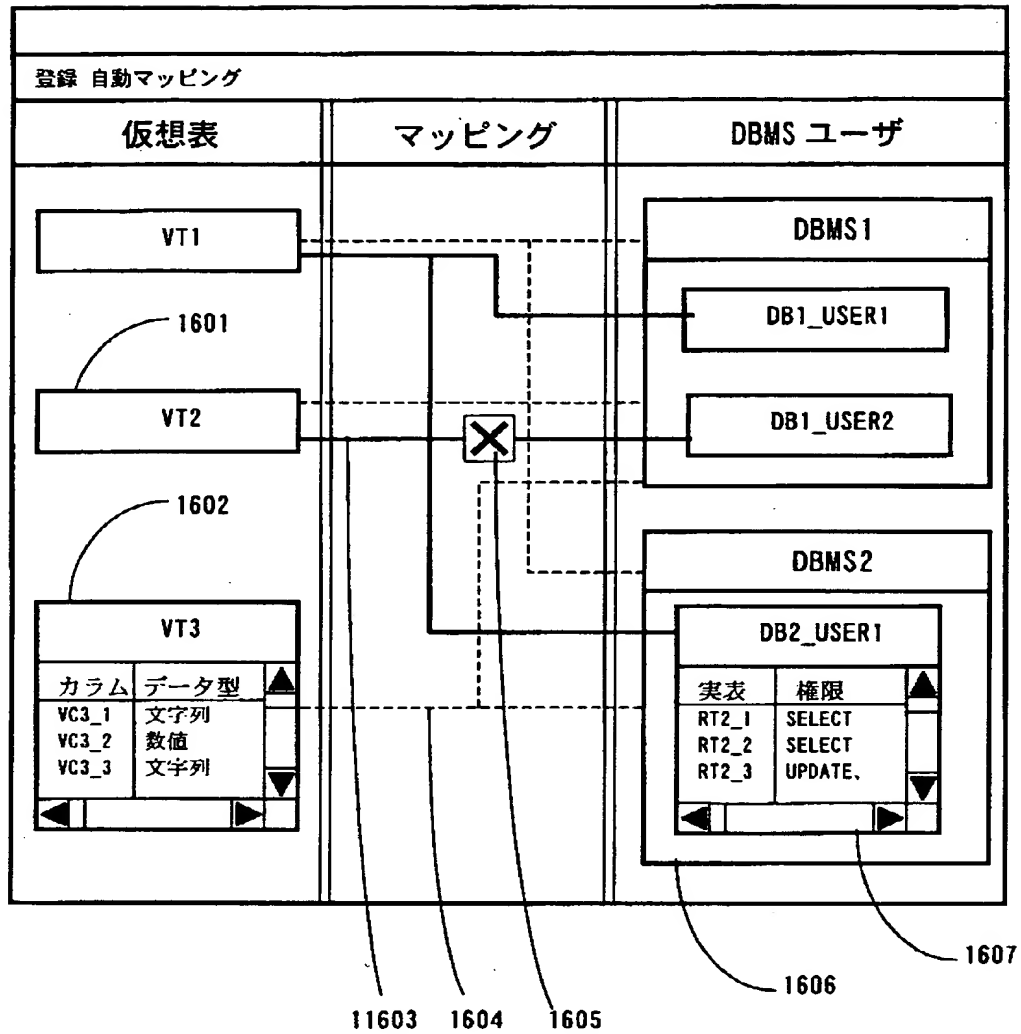
【図 1 5】

図 1 5



【図 1 6】

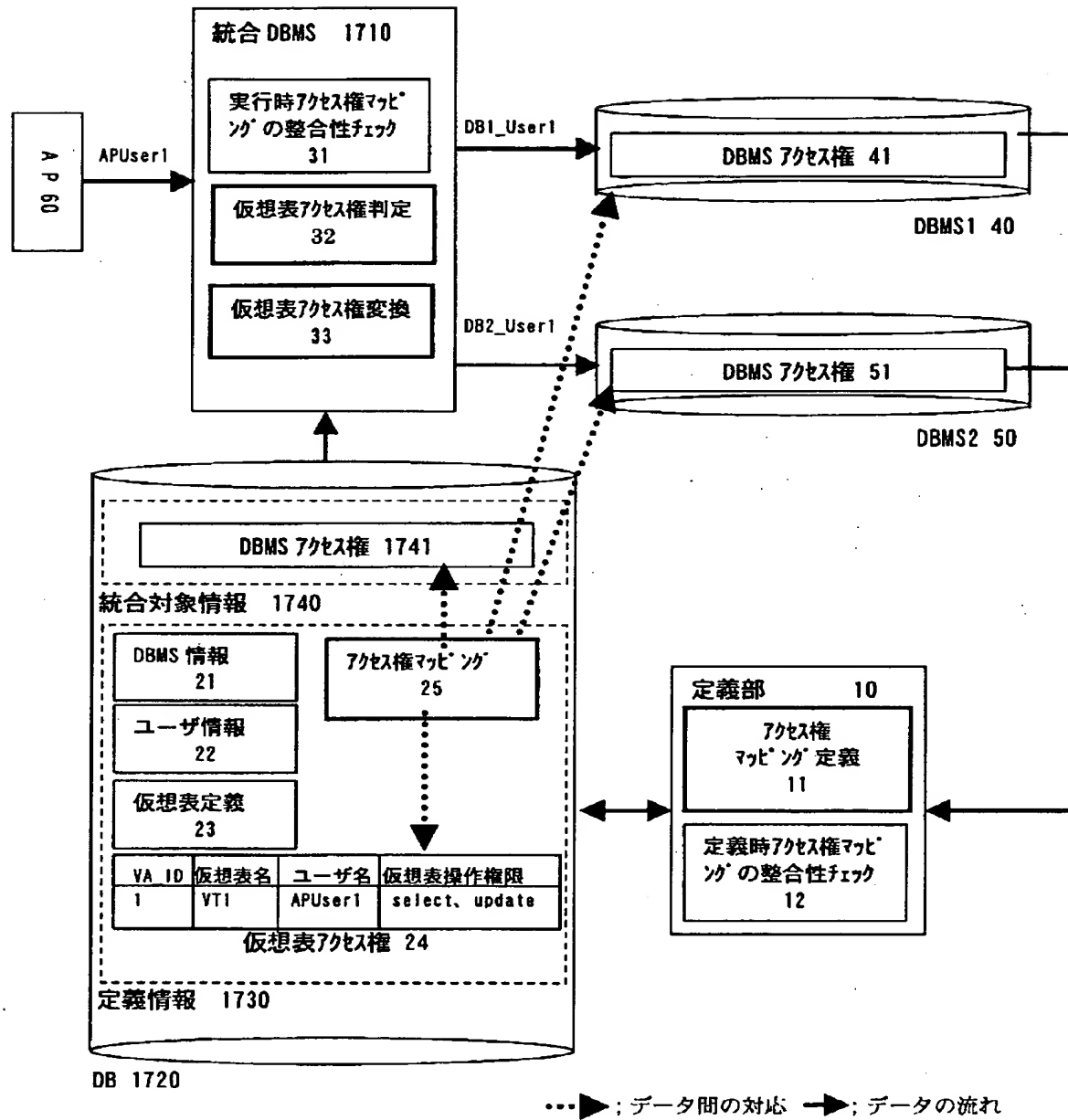
図 1 6





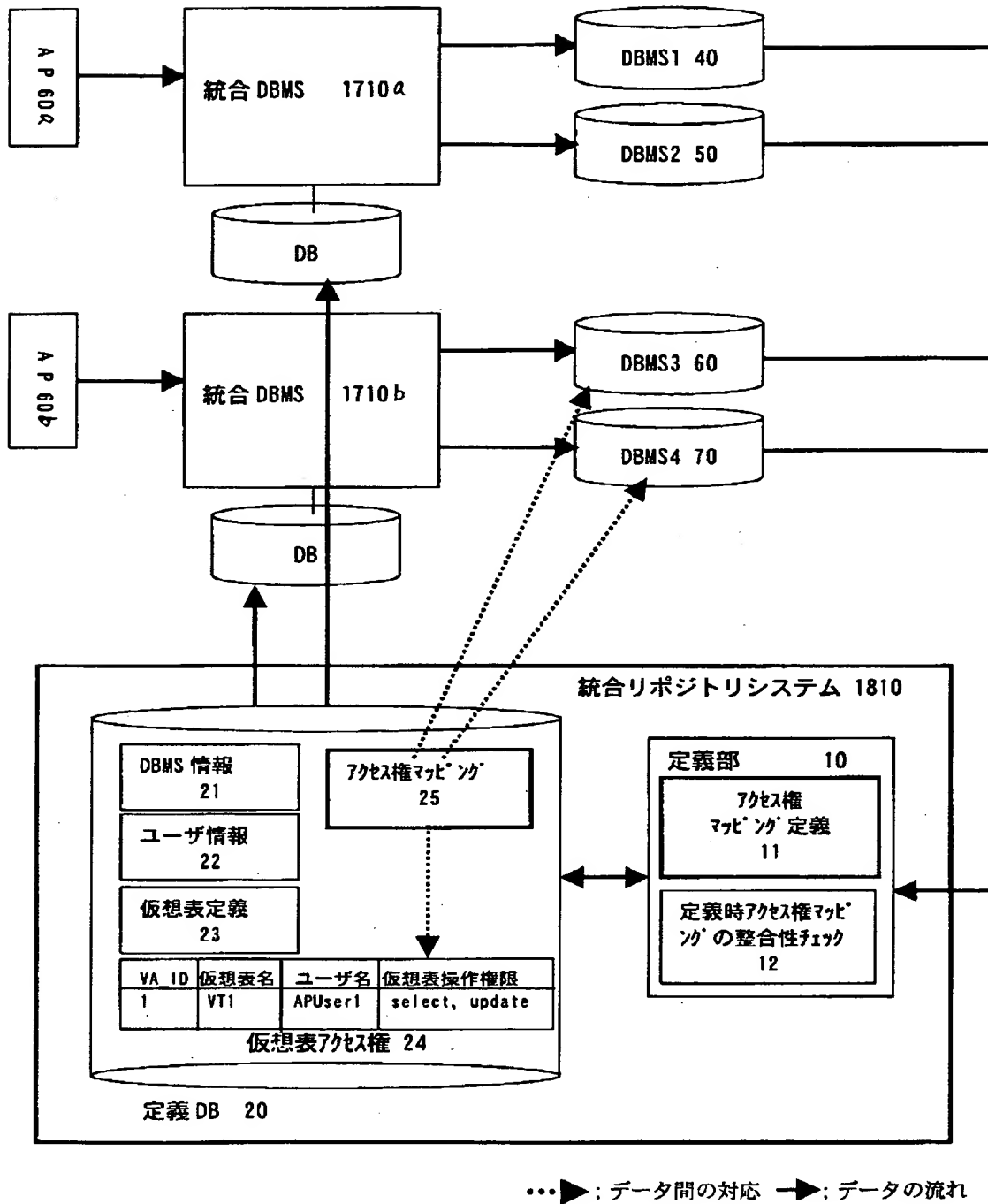
【図 1 7】

図 1 7



【図 18】

図 18



【書類名】 要約書

【要約】

【課題】

システム構築を行なう際、マルチデータベース統合システム上にて構築されるアプリケーション（AP）側と統合対象データベース管理システム（DBMS）側の管理組織が異なる場合がある。こうした場合、AP側と統合対象DBMS間の接続を行なうには、それぞれ独立した形態でシステム構築作業を進められることが望ましい。

【解決手段】

本発明では、仮想表アクセス権、スポークDBアクセス権、およびこれらに対応づけるアクセス権マッピング情報を定義する手段と管理する手段を設ける。そして上記情報の定義内容を用いることにより、仮想表レベルでのアクセス権判定を行なう処理部と、仮想表アクセス権からスポークDBアクセス権への変換処理部を設ける。本発明により、容易にAP側と統合対象DBMSのシステム構築作業を並列に行なうことができ、システム構築作業の負荷を低減する。

【選択図】 図 1

出 願 人 履 歴 情 報

識別番号 [000005108]

1. 変更年月日 1990年 8月31日  
[変更理由] 新規登録  
住 所 東京都千代田区神田駿河台4丁目6番地  
氏 名 株式会社日立製作所